X-603-71-249
PREPRINT

SGINDEX—AN OS/360 SYSTEM GENERATION CROSS-REFERENCE INDEX

C. WRANDLE BARTH

JUNE 1971

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-603-71-249

SGINDEX — AN OS/360 SYSTEM GENERATION CROSS-REFERENCE INDEX

C. Wrandle Barth

June 1971

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

CONTENTS

SGINDEX — AN OS/360 SYSTEM GENERATION CROSS-REFERENCE INDEX

INTRODUCTION

In almost all kinds of system programming under OS/360 it is necessary to extract information from the sysgen listing.* This is true whether one is modifying OS, applying PTF's, or writing subsystems which interface with OS. Finding needed information was a long and painstaking process of manually paging through the sysgen output. A need was present for an easy-to-use but comprehensive cross-reference index of the sysgen.

SGINDEX is a program which collects key data from the Stage II input for an OS/360 sysgen, sorts it, and prints a formatted listing of the index entries collected. Several optional features have been included in SGINDEX which can be requested via the EXEC card parameter.

SGINDEX uses a region of about 200K in an MVT environment, although this may be decreased by compiling with OPT=Ø. The program is written in PL/I and requires SYS1.PL1LIB for linkage editing. Since the OS sort package is used, SYS1.SORTLIB is required at run time. Using OPT=1, a large release 19 sysgen was run in slightly over 3 minutes on a 360/75J.

SGINDEX OUTPUT

The index entries collected are of the form:

ITEM LIB USE STEP PAGE CARD

ITEM is the actual index item extracted from the SYSGEN stage II input deck.

LIB is the library in which ITEM occurs, and is present on certain items only.

USE is a fixed term describing the environment in which the item was found.

STEP is the stepname of the step in which the item was found, PAGE is the page of the listing on which it appears, and CARD is the card number listed in columns 73-80.

---

*The sysgen listing is the output of the programs run to create a new operating system. The reader is assumed to be familiar with the generating of Operating System/360.

The following entries are extracted:

- The stepname of each SYSGEN step appears with the USE "STEPNAME".

- The program named in the "EXEC PGM=" parameter appears with the USE "EXEC".

- If the program is "ASMBLR", then:

  (1) The member listed on the "//SYSPUNCH DD" card appears with USE "SYSPUNCH" and with its dsname as LIB.

  (2) Each macro evoked appears with the USE "MACRO".

- If the program is "IEWL", then:

  (1) The load module(s) created appear with USE "LINK" and the SYSLMOD dsname as LIB.

  (2) Alias and entry names appear with USE "ALIAS" and "ENTRY" respectively and the SYSLMOD dsname as LIB.

  (3) Include card members appear as USE "INCLUDE" with the ddname operand as ' B.

  (4) Change card new-names appear as USE "CHANGE" with old-names as "LIB".

  (5) Replace card members appear as USE "REPLACE".

- If the program is "IEBCOPY", then:

  (1) Sending library name appears as USE "COPYFROM".

  (2) Moved member names appear as USE "COPYTO" with receiving library dsname as LIB.

  (3) Exclude card members appear as USE "EXCLUDE" with receiving library dsname as LIB.

The following JCL is required.

//---- JOB ----

JOB card as required by installation.

```
//STEPNAME EXEC PGM=SGINDEX,REGION=____,PARM=____
```

EXEC card; SGINDEX should run in about 220K. PARM is used to select run options—see discussion below.

```
//STEPLIB DD DSN=____,DISP=SHR
```

If SGINDEX is not in a LINKLIB, use a STEPLIB card to get it (or execute in a compile-and-go environment).

```
//SYSOUT DD SYSOUT=A (or "DUMMY")
```

Data set to receive SORT/MERGE messages.

```
//INDEX DD SYSOUT=A,DCB=(BLKSIZE=7260,LRECL=121,RECFM=FBA)
```

Data set to receive printed listing and index; blocksize may be varied.

```
//NEWDECK DD SYSOUT=B (or other data set)
```

Data set to receive modified copy of stage II deck. May be on disk or DUMMY also.

```
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=629)
```

Data set to receive PL/I error messages.

```
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
```

Standard SORT library.

```
//SORTWK01 DD UNIT=2314,SPACE=(CYL,3,,CONTIG)

//SORTWK02 DD UNIT=2314,SPACE=(CYL,3,,CONTIG),SEP=SORTWK01

//SORTWK03 DD UNIT=(2314,SEP=SORTWK01),SPACE=(CYL,3,,CONTIG),
//         SEP=SORTWK02

//SORTWK04 DD UNIT=(2314,SEP=SORTWK02),SPACE=(CYL,3,,CONTIG),
//         SEP=SORTWK01

//SORTWK05 DD UNIT=(2314,SEP=(SORTWK01,SORTWK03)),
//         SPACE=(CYL,3,,CONTIG),SEP=SORTWK02

//SORTWK06 DD UNIT=(2314,SEP=(SORTWK02,SORTWK04)),
//         SPACE=(CYL,3,,CONTIG),SEP=SORTWK01
```

Sort work data sets; these are listed only as guides which worked well at the author's installation. See SORT/MERGE manual for further discussion.

//SYSIN DD * (or point to data set)

Data set containing stage I output (stage II input) deck.

The format of the EXEC PGM=SGINDEX PARM is:

PARM=(LLL,RRRRRRR,OOOOOO,NNNNN,III...I,SSS...S)

LLL is an integer such that 9 < LLL < 32767. The listing and index will be printed with LLL lines per page (including headings and blank lines). If omitted or invalid, LLL will default to 60.

RRRRRRR is the release number. If present it will appear at the top of each page of the listing and index. The first two characters, if numeric, will appear as the two low-order digits of the cards listed and punched on NEWDECK, unless NNNNN is "NOSEQ".

OOOOOO is "OBJDEL" or omitted. If present it indicates that any object module cards in the input deck (SYSIN) are to be deleted. This allows one to include in his stage I input cards of the form "SYSGEN-MACRO-NAME EQU *" which will cause the assembler to provide in its standard cross reference listing an index of the stage I macros by line number. However, the assembler will also output an ESD and an END card with the stage I output; "OBJDEL" will delete these.

NNNNN is "NOSEQ" or omitted. If present, columns 73-80 of the input will be output to the listing and to NEWDECK unaltered. If omitted, sequence numbers will be generated beginning with 10000+RR (where RR is the first two digits of RRRRRRR, above, if numeric; else RR=00), and incrementing by 10000.

III...I is a string of characters which will appear on listing and index page headings. It may be up to 16 characters long.

SSS...S is a string of up to 57 characters. If present the first card image of every "//SYSPRINT DD" statement will have its operand field replaced with the character string entered. As of this writing, this string is always the space allocation; this allows the user to replace the SYSGEN space allocation for SYSPRINT. If omitted, no SYSPRINT modifications are made.

The parameters are positional, with omitted parameters indicated by comma only, and with trailing commas optional. Spaces will not delimit any parameter to SGINDEX, but if any parameter contains spaces or any other special character, it must be enclosed in primes ('). For example:

```
//INDEX19 EXEC PGM=SGINDEX,REGION=220K,PARM=(,'19.6',OBJDEL,
//          ,'360 75J','SPACE=(121,(300,60)),  *** REPLACED ***')
```

requests 60 lines per page, pages headed with "RELEASE 19.6" and "360 75J", object cards deleted, cards resequenced beginning with 00010019, and SYSPRINT dd cards replaced with

```
//SYSPRINT DD SPACE=(121,(300,60)),  *** REPLACED ***
```

Since the release number contained a period, it was placed within primes. Note that if SSS...S is included, it should include a trailing comma to continue the SYSPRINT cards. Commas in any other parameter are treated as delimiters. This means that the second card above could have been specified as

```
//          ',360 75J,SPACE=(121,(300,60)),  *** REPLACED ***')
```

with identical results.

INSTALLING SGINDEX

Normal distribution medium for SGINDEX is a 9-track, 800 bpi unlabelled distribution tape reel (DTR). It contains two data sets—the PL/1 source of SGINDEX and a compiled object module. Each data set contains 80-byte records, blocked to 2000 bytes.

When compiling SGINDEX, include the parameter SM=(2,72,1) in the compiler PARM field.

SGINDEX includes a PL/1-coded procedure "VERIFY" which is a built-in function in PL/1 version 5. If compiled on a version 5 compiler, the procedure and its declaration should be removed to allow the built-in version to be utilized.

SGINDEX PROGRAM LOGIC

Internally, SGINDEX is written utilizing PL/I multitasking. The mother task scans the input deck for the needed index entries. The daughter is attached

initially and it invokes the sort routine and points to the procedure INDEX_ ENTRY_SENDER as the routine to provide sort records. When it is entered, it waits for the mother task to produce a record whereupon it copies the record into its area, posts the mother to continue, and returns the record to the sort routine. All of this allows the mother's OUTPUT_SORT_KEY procedure to be called many levels down from many places and still be able to pass the record tc the sorter and return to its caller. Each procedure in SGINDEX is listed below with its function.

SYSGEN_INDEX_CREATION: This is the main block of the program. It processes the EXEC PGM=SGINDEX card parm, setting switches for the various options requested and saving the fields specified. The data sets are opened. SORT_INTERFACE is attached as a subtask. A loop is then entered to process each step of the stage II input deck. READER is used to fetch the next EXEC card. The stepname and program name are passed as entries to the sorter. LINK_PROCESS, ASMBLR_PROCESS, or COPY_PROCESS are evoked if the program is IEWL, ASMBLR, or IEBCOPY. When the input ends, RECORD_ GENERATED is marked complete with a status of 4 and a WAIT occurs on INDEX_PRINTED which is posted when the subtask ends. SGINDEX then exits.

LINK_PROCESS: This block extracts the SYSLMOD library and member names, outputs index entries for most linkage editor control cards, and exits at the end of the control card data set.

ASMBLR_PROCESS: This block extracts the SYSPUNCH dsname and member to output as an index entry, loops to output each macro instruction as an index entry, and exits at end of control card data set.

COPY_PROCESS: This block scans the dsnames of the SYSUT1 and SYSUT2 cards, outputs index entries for each member copied, cleans up, and exits.

SYSUT1_SET: This short block outputs an index entry for USE "COPYFROM" whether item came from SYSUT1 dd card or INDD= in copy control card.

OUTPUT_ALL_KEYS: This block gets fields from the string and outputs index entries for those items, looping as long as the fields are delimited with commas.

OUTPUT_SORT_KEYS: This short block passes the index entry to SORT_INTERFACE and keeps count.

READER: This block reads SYSGEN stage II records, inserting sequence numbers and replacing SYSPRINT cards if requested, and echoing them on the listing. This is repeated until a card of the type requested is read. Control is

then returned with the card's columns 1-71 in CARD, column 72 in CONTINUATION_COLUMN, EXEC_CARD set to '1'B if an exec card was read, and JCL set to '1'B if any "//" card was read.

FIELD_SCAN: This block scans for a field within the character string STRING. STRING normally contains part (or all) of the card image currently being processed. If SPACE_BYPASS is set, non-blanks before the first blank and then blanks up to the next non-blank are ignored. If EQUAL_SCAN_REQUEST is set, more characters are discarded until an equal sign is found. The field returned is the string beginning at the place indicated above, and terminating with a comma, space, or open or close parenthesis. (If the string begins with any of these delimiters, they are discarded.) The resulting string is returned in FIELD. The delimiting character is returned in DELIMITER. Characters processed are removed from STRING up to but excluding the delimiter.

JCL_SCAN: This block will search through DD cards until the card whose name (or partial name) is stored in DDNAME is found. The DSNAME= parameter is then found and the second level of the data set name is extracted and returned in DSNAME. If a member name is present, it is returned in MEMBER.

SORT_INTERFACE: This procedure is attached as a subtask. It evokes the sort routine which enters INDEX_ENTRY_SENDER to fetch each of the sort records collected by the mother task. Sort then enters INDEX_PRINTER to print each entry. When sort returns we finish off the printout and exit.

INDEX_ENTRY_SENDER: This block is entered as sort exit point 15. It waits for the mother task to complete the next record. When the record is copied out the mother is signaled to process another and we return the record to SORT/MERGE. When all records have been passed, preparations are made to print and the final return is made.

INDEX_PRINTER: This block is called by the sort routine as exit point 35. It prints the page in a two-column format. Left-hand entries are saved until the corresponding right-hand entries are received, at which time both are printed.

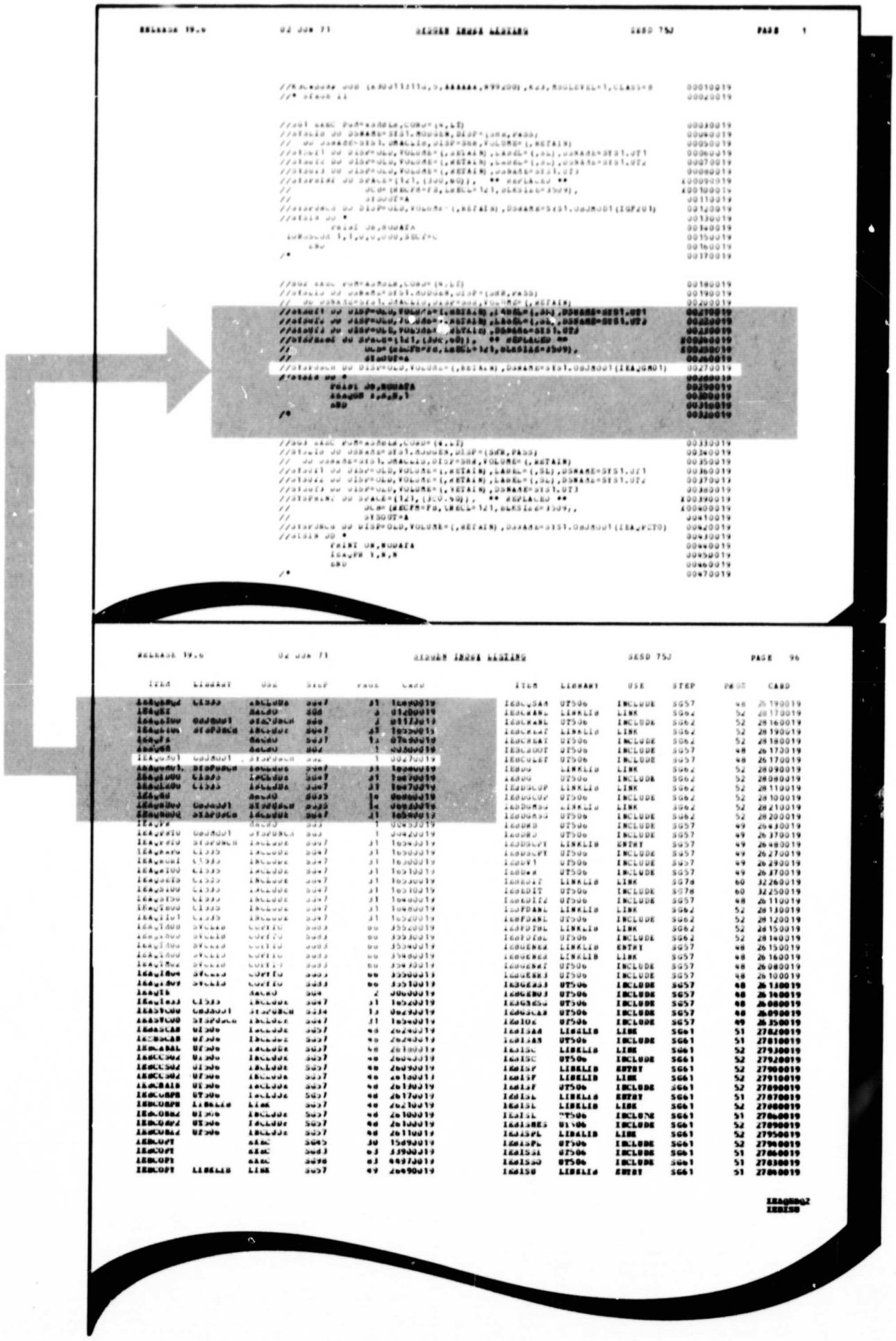
PRINT_THUMB_TAB: This block prints the thumb tabs for the first and last index items on the page at the bottom of the page.

VERIFY: This block simulates the "VERIFY" function which is a built-in function with PL/I version 5 (release 19). It is included here for running under release 18 and should be removed later to use the built-in version.

APPENDIX A

SAMPLE OUTPUT

PRECEDING PAGE BLANK NOT FILMED



APPENDIX B

SOURCE LISTING

```
VERSION 4.3                                   05/360 PL/I COMPILER (F)

PL/I F COMPILER OPTIONS SPECIFIED ARE AS FOLLOWS--

     O=1,SR=(2,72,1),LIST,SZ,ND

THE COMPLETE LIST OF OPTIONS USED DURING THIS COMPILATION IS--   EBCDIC
                                                                CHAR60
                                                              NOMACRO
                                                                SOURCE2
                                                              NOMACDCK
                                                                COMP
                                                                SOURCE
                                                                ATR
                                                                XREF
                                                                ATTRIB
                                                                LIST
                                                                LOAD
                                                              NODECK
                                                                FLAGE
                                                                STMT
                                                                SIZE=0196756
                                                                LINECNT=058
                                                                OPT=01
                                                                SORMGIN=(002,072,001)
                                                              NOXREF
                                                                MAT
                                                                OPLIST

*OPTIONS IN EFFECT*     EBCDIC,CHAR60,NOMACRO,SOURCE2,NOMACDCK,COMP,SOURCE,ATR,XREF,ATTRIB,LIST,LOAD,
*OPTIONS IN EFFECT*     NODECK,FLAGE,STMT,SIZE=0196756,LINECNT=058,OPT=01,SORMGIN=(002,072,001),NOXREF,
*OPTIONS IN EFFECT*     MAT,OPLIST
```

```
/* *                    "S G I N D E X"                    * */00001000

STMT LEVEL NEST
/* *                    "S G I N D E X"                    * */00001000
/* *                                                       * */00002000
/* *                 SYSGEN INDEX COLLECTOR                * */00003000
/* *              WRITTEN BY C. BRANDLE SMITH              * */00004000
/* *              GODDARD SPACE FLIGHT CENTER              * */00005000
/* *                   22 FEBRUARY 1971                    * */00006000
/* *                                                       * */00007000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00008000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00009000

                                                             00010000

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00011000
/*                                                         */00012000
/*     "SGINDEX" IS A PROGRAM WHICH COLLECTS KEY DATA FROM THE */00013000
/*  STAGE II INPUT FOR AN OS/360 SYSGEN, SORTS IT, AND PRINTS A */00014000
/*  FORMATTED LISTING OF THE INDEX ENTRIES COLLECTED.  SEVERAL */00015000
/*  OPTIONAL FEATURES HAVE BEEN INCLUDED IN "SGINDEX" WHICH CAN */00016000
/*  BE REQUESTED VIA THE EXEC CARD PARAMETER.              */00017000
/*                                                         */00018000
/*     THE INDEX ENTRIES COLLECTED ARE OF THE FORM:        */00019000
/*                                                         */00020000
/*           ITEM  LIB  USE  STEP  PAGE  CARD              */00021000
/*                                                         */00022000
/*  "ITEM" IS THE ACTUAL INDEX ITEM EXTRACTED FROM THE SYSGEN STAGE */00023000
/*  II INPUT DECK.  "LIB" IS THE LIBRARY IN WHICH "ITEM" OCCURRS, */00024000
/*  AND IS PRESENT ON CERTAIN ITEMS ONLY.  "USE" IS A FIXED TERM */00025000
/*  DESCRIBING THE ENVIRONMENT IN WHICH THE ITEM WAS FOUND.  "STEP" */00026000
/*  IS THE STEPNAME OF THE STEP IN WHICH THE ITEM WAS FOUND, "PAGE" */00027000
/*  IS THE PAGE OF THE LISTING ON WHICH IT APPEARS, AND "CARD" IS */00028000
/*  THE CARD NUMBER LISTED IN COLUMNS 73-80.               */00029000
/*                                                         */00030000
/*  THE FOLLOWING ENTRIES ARE EXTRACTED:                   */00031000
/*                                                         */00032000
/*     - THE STEPNAME OF EACH SYSGEN STEP APPEARS WITH THE */00033000
/*       USE "STEPNAME".                                   */00034000
/*     - THE PROGRAM NAMED IN THE "EXEC PGM=" PARAMETER    */00035000
/*       APPEARS WITH THE USE "EXEC".                      */00036000
/*     - IF THE PROGRAM IS "ASMBLR", THEN:                 */00037000
/*          1) THE MEMBER LISTED ON THE "//SYSPUNCH DD"    */00038000
/*             CARD APPEARS WITH THE USE "SYSPUNCH" AND    */00039000
/*             WITH ITS DSNAME AS LIB.                     */00040000
/*          2) EACH MACRO INVOKED APPEARS WITH THE         */00041000
/*             USE "MACRO".                                */00042000
/*     - IF THE PROGRAM IS "IEWL", THEN:                   */00043000
/*          1) THE LOAD MODULE(S) CREATED APPEAR WITH      */00044000
/*             USE "LINK" AND THE SYSLMOD DSNAME AS LIB.   */00045000
/*          2) ALIAS AND ENTRY NAMES APPEAR WITH USE       */00046000
/*             "ALIAS" AND "ENTRY" RESPECTIVELY AND        */00047000
/*                                                         */00048000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00049000
```

```
/* *                    " S G I N D E X "                        * */00001000

STMT LEVEL NEST

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00050000
/*                                                                 */00051000
/*            THE SYSLMOD DDNAME AS LIB.                           */00052000
/*         3) INCLUDE CARD MEMBERS APPEAR AS USE                   */00053000
/*            "INCLUDE" WITH THE DDNAME OPERAND AS LIB.            */00054000
/*         4) CHANGE CARD NEW-NAMES APPEAR AS USE                  */00055000
/*            "CHANGE" WITH OLD-NAMES AS "LIB".                    */00056000
/*         5) REPLACE CARD MEMBERS APPEAR AS USE                   */00057000
/*            "REPLACE".                                           */00058000
/*     - IF THE PROGRAM IS "IEBCOPY", THEN:                        */00059000
/*         1) SENDING LIBRARY NAME APPEARS AS USE                  */00060000
/*            "COPYFROM".                                          */00061000
/*         2) MOVED MEMBER NAMES APPEAR AS USE                     */00062000
/*            "COPYTO" WITH RECEIVING LIBRARY                      */00063000
/*            DDNAME AS LIB.                                       */00064000
/*         3) EXCLUDE CARD MEMBERS APPEAR AS USE                   */00065000
/*            "EXCLUDE" WITH RECEIVING LIBRARY                     */00066000
/*            DDNAME AS LIB.                                       */00067000
/*                                                                 */00068000
/*     THE FOLLOWING JCL IS REQUIRED.                              */00069000
/*                                                                 */00070000
/* //---- JOB ----                                                 */00071000
/*                                  JOB CARD AS REQUIRED BY INSTALLA-*/00072000
/*                                       TION.                     */00073000
/* //STEPNAME EXEC PGM=SGINDEX,REGION=___K,PARM='___'             */00074000
/*                                  EXEC CARD;  SGINDEX SHOULD RUN  */00075000
/*                                       IN ABOUT 220K.  PARM IS USED*/00076000
/*                                       TO SELECT RUN OPTIONS---SEE*/00077000
/*                                       DISCUSSION BELOW.         */00078000
/* //STEPLIB DD DSN=___,DISP=SHR                                  */00079000
/*                                  IF SGINDEX IS NOT IN A LINKLIB, */00080000
/*                                       USE A STEPLIB CARD TO     */00081000
/*                                       GET IT (OR EXECUTE IN A   */00082000
/*                                       COMPILE-AND-GO ENVIRONMENT).*/00083000
/* //SYSOUT DD SYSOUT=A (OR "DUMMY")                              */00084000
/*                                  DATA SET TO RECEIVE SORT/MERGE  */00085000
/*                                       MESSAGES.                 */00086000
/* //INDEX DD SYSOUT=A,DCB=(BLKSIZE=7280,LRECL=121,RECFM=FBA)      */00087000
/*                                  DATA SET TO RECEIVE PRINTED LISTING*/00088000
/*                                       AND INDEX;  BLOCKSIZE MAY  */00089000
/*                                       BE VARIED.                */00090000
/* //NEWDECK DD SYSOUT=B (OR OTHER DATA SET)                      */00091000
/*                                  DATA SET TO RECEIVE MODIFIED    */00092000
/*                                       COPY OF STAGE II DECK.    */00093000
/*                                       MAY BE ON DISK OR DUMMY ALSO.*/00094000
/* //SYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=629)   */00095000
/*                                       DATA SET TO RECEIVE PL/I ERROR*/00096000
/*                                                                 */00097000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00098000
```

```
/* *                    " S G I N D E X "                        * */00001000

STMT LEVEL NEST

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00099000
/*                                                                 */00100000
/*                                       MESSAGES.                 */00101000
/* //SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR                         */00102000
/*                                       STANDARD SORT LIBRARY.    */00103000
/* //SORTWK01 DD UNIT=2314,SPACE=(CYL,3,,CONTIG)                  */00104000
/* //SORTWK02 DD UNIT=2314,SPACE=(CYL,3,,CONTIG),SEP=SORTWK01     */00105000
/* //SORTWK03 DD UNIT=(2314,SEP=SORTWK01),SPACE=(CYL,3,,CONTIG),  */00106000
/* //            SEP=SORTWK02                                     */00107000
/* //SORTWK04 DD UNIT=(2314,SEP=SORTWK02),SPACE=(CYL,3,,CONTIG),  */00108000
/* //            SEP=SORTWK01                                     */00109000
/* //SORTWK05 DD UNIT=(2314,SEP=(SORTWK01,SORTWK03)),             */00110000
/* //            SPACE=(CYL,3,,CONTIG),SEP=SORTWK02               */00111000
/* //SORTWK06 DD UNIT=(2314,SEP=(SORTWK02,SORTWK04)),             */00112000
/* //            SPACE=(CYL,3,,CONTIG),SEP=SORTWK01               */00113000
/*                                  SORT WORK DATA SETS;  THESE ARE */00114000
/*                                       LISTED ONLY AS GUIDES WHICH*/00115000
/*                                       WORKED WELL AT THE AUTHOR'S*/00116000
/*                                       INSTALLATION.  SEE SORT/MERGE*/00117000
/*                                       MANUAL FOR FURTHER DISCUS- */00118000
/*                                       SION.                     */00119000
/* //SYSIN DD * (OR POINT TO DATA SET)                            */00120000
/*                                  DATA SET CONTAINING STAGE I OUTPUT*/00121000
/*                                       (STAGE II INPUT) DECK.    */00122000
/*                                                                 */00123000
/*     THE FORMAT OF THE EXEC PGM=SGINDEX PARM IS:                 */00124000
/*                                                                 */00125000
/*     PARM='LLL,NNNNNNNN,OOOOOO,MMMMM,III...I,SSS...S'            */00126000
/*                                                                 */00127000
/*    LLL IS A INTEGER SUCH THAT 9 < LLL < 32767.  THE LISTING AND */00128000
/*     INDEX WILL BE PRINTED WITH LLL LINES PER PAGE (INCLUDING    */00129000
/*     HEADINGS AND BLANK LINES).  IF OMITTED OR INVALID, LLL WILL */00130000
/*     DEFAULT TO 60.                                              */00131000
/*    NNNNNNNN IS THE RELEASE NUMBER.  IF PRESENT IT WILL APPEAR AT THE*/00132000
/*     TOP OF EACH PAGE OF THE LISTING AND INDEX.  THE FIRST TWO    */00133000
/*     CHARACTERS, IN NUMERIC, WILL APPEAR AS THE TWO LOW-ORDER     */00134000
/*     DIGITS OF THE CARDS LISTED AND PUNCHED ON "NEWDECK", UNLESS  */00135000
/*     MMMMM IS "NOSEQ".                                           */00136000
/*    OOOOOO IS "OBJDEL" OR OMITTED.  IF PRESENT IT INDICATES THAT */00137000
/*     ANY OBJECT MODULE CARDS IN THE INPUT DECK ("SYSIN") ARE TO   */00138000
/*     BE DELETED.  THIS ALLOWS ONE TO INCLUDE IN HIS STAGE I INPUT */00139000
/*     CARDS OF THE FORM "SYSGEN-MACRO-NAME  EQU  *" WHICH WILL     */00140000
/*     CAUSE THE ASSEMBLER TO PROVIDE IN ITS STANDARD CROSS REFER-  */00141000
/*     ENCE LISTING AN INDEX OF THE STAGE I MACROS BY PAGE NUMBER.  */00142000
/*     HOWEVER THE ASSEMBLER WILL ALSO OUTPUT AN "ESD" AND AN "END" */00143000
/*     CARD WITH THE STAGE I OUTPUT;  "OBJDEL" WILL DELETE THESE.   */00144000
/*    MMMMM IS "NOSEQ" OR OMITTED.  IF PRESENT, COLUMNS 73-80 OF THE*/00145000
/*     INPUT WILL BE OUTPUT TO THE LISTING AND TO "NEWDECK"         */00146000
/*     UNALTERED.  IF OMITTED, SEQUENCE NUMBERS WILL BE GENERATED   */00147000
/*                                                                 */00148000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00149000
```

```
/* *                    " S G I N D E X "                      * */00001000

STMT LEVEL NEST

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00150000
/*                                                               */00151000
/*     BEGINNING WITH 1CCCCCNN (WHERE NN IS THE FIRST TWO DIGITS */00152000
/*     OF NNNNNN, ABOVE, IF NUMERIC;  ELSE NN = 00), AND INCRE-  */00153000
/*     MENTING BY 10000.                                         */00154000
/*  111...1 IS A STRING OF CHARACTERS WHICH WILL APPEAR ON LISTING */00155000
/*     AND INDEX PAGE HEADINGS.  IT MAY BE UP TO 16 CHARACTERS LONG.*/00156000
/*  333...3 IS A STRING OF UP TO 57 CHARACTERS.  IF PRESENT       */00157000
/*     THE FIRST CARD IMAGE OF EVERY "//SYSPRINT DD" STATEMENT   */00158000
/*     WILL HAVE ITS OPERAND FIELD REPLACED WITH THE CHARACTER   */00159000
/*     STRING ENTERED.  AS OF THIS WRITING, THIS STRING IS ALWAYS */00160000
/*     THE SPACE ALLOCATION;  THIS ALLOWS THE USER TO REPLACE THE */00161000
/*     STOCKH SPACE ALLOCATION FOR SYSPRINT.  IF OMITTED, NO     */00162000
/*     SYSPRINT MODIFICATIONS ARE MADE.                          */00163000
/*                                                               */00164000
/*     THE PARAMETERS ARE POSITIONAL, WITH OMITTED PARAMETERS    */00165000
/*  INDICATED BY COMMA ONLY, WITH TRAILING COMMAS OPTIONAL.  SPACES */00166000
/*  WILL NOT DELIMIT ANY PARAMETER TO "SGINDEX", BUT IF ANY PARAMETER*/00167000
/*  CONTAINS SPACES OR ANY OTHER SPECIAL CHARACTER, IT MUST BE    */00168000
/*  ENCLOSED IN PRIMES (').  FOR EXAMPLE:                        */00169000
/*                                                               */00170000
/*  //INDEX19  EXEC  PGM=SGINDEX,REGION=220K,PARM=(,'19.6',OBJDEL, */00171000
/*  //         ,'360 75J','SPACE=(121,(300,60))',  *** REPLACED ***') */00172000
/*                                                               */00173000
/*  REQUESTS 60 LINES PER PAGE, PAGES HEADED WITH "RELEASE 19.6" AND */00174000
/*  "360 75J", OBJECT CARDS DELETED, CARDS RESEQUENCED BEGINNING */00175000
/*  WITH 00010019, AND SUSPRINT DD CARDS REPLACED WITH           */00176000
/*                                                               */00177000
/*  //SYSPRINT DD SPACE=(121,(300,60)),  *** REPLACED ***        */00178000
/*                                                               */00179000
/*  SINCE THE RELEASE NUMBER CONTAINED A PERIOD, IT WAS PLACED WITHIN*/00180000
/*  PRIMES.  NOTE THAT IF 333...3 IS INCLUDED, IT SHOULD INCLUDE A */00181000
/*  TRAILING COMMA TO CONTINUE THE SYSPRINT CARDS.  COMMAS IN ANY */00182000
/*  OTHER PARAMETER ARE TREATED AS DELIMITERS.  THIS MEANS THAT THE */00183000
/*                                                               */00184000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00185000
```

```
/* *                    " S G I N D E X "                      * */00001000

STMT LEVEL NEST

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00186000
/*                                                               */00187000
/*  SECOND CARD ABOVE COULD HAVE BEEN SPECIFIED AS               */00188000
/*                                                               */00189000
/*  //         ',360 75J,SPACE=(121,(300,60)),  *** REPLACED ***') */00190000
/*                                                               */00191000
/*  WITH IDENTICAL RESULTS.                                      */00192000
/*                                                               */00193000
/*     INTERNALLY, "SGINDEX" IS WRITTEN UTILIZING PL/I MULTI-    */00194000
/*  TASKING.  THE MOTHER TASK SCANS THE INPUT DECK FOR THE NEEDED */00195000
/*  INDEX ENTRIES.  THE DAUGHTER IS ATTACHED INITIALLY WHICH     */00196000
/*  INVOKES THE SORT ROUTINE AND POINTS TO THE PROCEDURE "INDEX_ */00197000
/*  ENTRY_SENDER" AS THE ROUTINE TO PROVIDE SORT RECORDS.  WHEN IT */00198000
/*  IS ENTERED, IT WAITS FOR THE MOTHER TASK TO PRODUCE A RECORD */00199000
/*  WHEREUPON IT COPIES THE RECORD INTO ITS AREA, POSTS THE MOTHER */00200000
/*  TO CONTINUE, AND RETURNS THE RECORD TO THE SORT ROUTINE.     */00201000
/*  ALL OF THIS ALLOWS THE MOTHER'S "OUTPUT_SORT_KEY" PROCEDURE  */00202000
/*  TO BE CALLED MANY LEVELS DOWN FROM MANY PLACES AND STILL BE  */00203000
/*  ABLE TO PASS THE RECORD TO THE SORTER AND RETURN TO ITS CALLER. */00204000
/*                                                               */00205000
/*     IF COMPILED ON A PL/I VERSION 5 COMPILER, THE PROCEDURE   */00206000
/*  FUNCTION "VERIFY" AND ITS DECLARATION SHOULD BE REMOVED AND  */00207000
/*  THE BUILT-IN FUNCTION UTILITIZED.                            */00208000
/*                                                               */00209000
/*     WHEN COMPILING "SGINDEX", USE PARM='SH=(2,71,1)'.         */00210000
/*                                                               */00211000
/*                                                  -CWB-        */00212000
/*                                                               */00213000
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */00214000
```

```
      /* *                    " S G I N D E X "                    * */00001000

DCL LEVEL NEST

  1             SGINDEX:  PROCEDURE (PARM) OPTIONS (MAIN, TASK);                  00215000

  2     1       FILES:  DECLARE                                                   00216000
                          SYSIN STREAM INPUT, /* STAGE II INPUT DECK             */00217000
                          PRINTER STREAM PRINT OUTPUT,                           00218000
                                              /* LISTING OF STAGE II DECK;      */00219000
                                              /*    DDNAME IS "INDEX".          */00220000
                          SENDECK STREAM OUTPUT;                                 00221000
                                              /* COPY OF DECK (WITH "SYSPRINT"  */00222000
                                              /*    CARDS REPLACED AND SEQUENCE */00223000
                                              /*    NUMBERED, IF REQUESTED).    */00224000

  3     1       BIT_STRINGS:                                                      00225000
                    DECLARE                                                      00226000
                          (ALL_DONE,          /* SYSIN END-OF-FILE INDICATOR.   */00227000
                          ERROR_OCCURRED,     /* INDICATES "FIELD_SCAN" FOUND   */00228000
                                              /*    A BLANK CARD, IMPROPER DE-  */00229000
                                              /*    LIMITER, ETC.               */00230000
                          SPACE_BYPASS,       /* INDICATES "FIELD_SCAN" SHOULD  */00231000
                                              /*    START SCAN WITH FIRST NON-  */00232000
                                              /*    BLANK AFTER FIRST BLANK IN  */00233000
                                              /*    "STRING".                   */00234000
                          EQUAL_SCAN_REQUEST, /* INDICATES "FIELD_SCAN" SHOULD  */00235000
                                              /*    (AFTER "SPACE_BYPASS", IF   */00236000
                                              /*    REQUESTED) SCAN FOR AN EQUAL*/00237000
                                              /*    SIGN BEFORE SELECTING FIELD.*/00238000
                          SELECT_MEMBER_SCAN, /* INDICATES SCAN OF "FIELCOPY" PRO-*/00239000
                                              /*    GRAM "SELECT" CARD MEMBERS IS*/00240000
                                              /*    BEING MADE.                 */00241000
                          OBJECT_CARD_DELETION_REQUESTED,                        00242000
                                              /* INDICATES ANY OBJECT CARDS READ*/00243000
                                              /*    FROM "SYSIN" SHOULD BE      */00244000
                                              /*    DELETED.                    */00245000
                          MODIFY_SYSPRINT_REQUESTED,                             00246000
                                              /* INDICATES THE FIRST CARD OF    */00247000
                                              /*    INCOMING "//SYSPRINT DD "   */00248000
                                              /*    STATEMENTS SHOULD HAVE THEIR*/00249000
                                              /*    OPERANDS REPLACED BY THE LAST*/00250000
                                              /*    PARM OPERAND OF THE "EXEC   */00251000
                                              /*    PGM=SGINDEX" CARD.          */00252000
                          SEQUENCE_SUPPRESS,  /* INDICATES THAT COLUMN 73-80    */00253000
                                              /*    OF INPUT DECK SHOULD NOT    */00254000
                                              /*    BE ALTERED;  SET IF FOURTH  */00255000
                                              /*    "EXEC PGM=SGINDEX" PARM IS  */00256000
                                              /*    "NOSEQ".                    */00257000
                          READ_TO_EXEC,       /* INDICATES "READER" SHOULD BYPASS*/00258000
                                              /*    ALL CARDS UNTIL NEXT EXEC   */00259000
                                              /*    CARD IS FOUND.              */00260000
                          READ_TO_NON_JCL,    /* INDICATES "READER" SHOULD BYPASS*/00261000
                                              /*    ALL JCL CARDS AND RETURN    */00262000

      /* *                    " S G I N D E X "                    * */00001000

DCL LEVEL NEST

                                              /*    NEXT NON-JCL CARD.          */00263000
                          READ_TO_NON_CONTIN, /* INDICATES "READER" SHOULD BYPASS*/00264000
                                              /*    CONTINUATION CARDS.         */00265000
                          JCL,                /* INDICATES LAST CARD READ IS A  */00266000
                                              /*    JCL ("//") CARD.            */00267000
                          EXEC_CARD)          /* INDICATES LAST CARD READ IS AN */00268000
                                              /*    EXEC CARD.                  */00269000
                        STATIC BIT (1) INITIAL (0B);                             00270000

  4     1       CHARACTER_STRINGS:                                                00271000
                    DECLARE                                                      00272000
                          PARM CHAR (100) VARYING,                               00273000
                                              /* "EXEC PGM=SGINDEX" PARM FIELD. */00274000
                         (RELEASE_INFO CHAR (15) INITIAL ('          '),         00275000
                                              /* CONTAINS EITHER BLANKS OR      */00276000
                                              /*    "RELEASE _____" FOR         */00277000
                                              /*    HEADING.                    */00278000
                          LIST_ID CHAR (16) INITIAL ('                '),       00279000
                                              /* CONTAINS EITHER BLANKS OR      */00280000
                                              /*    USER ID FOR HEADING.        */00281000
                          STRING CHAR (100) VARYING,                             00282000
                                              /* CONTAINS THAT PART OF THE      */00283000
                                              /*    CURRENT CARD BEING EXAMINED.*/00284000
                          SCAN_STRING CHAR (100) VARYING,                        00285000
                                              /* CONTAINS STRING (ABOVE) PLUS   */00286000
                                              /*    EXTRA CHARACTERS TO LIMIT   */00287000
                                              /*    SCAN DURING "FIELD_SCAN".   */00288000
                          FIELD CHAR (8),     /* FIELD FOUND BY "FIELD_SCAN".   */00289000
                          PREV_CONTIN CHAR (1) INITIAL (' '),                    00290000
                                              /* CONTINUATION CHARACTER SAVED   */00291000
                                              /*    FROM PREVIOUS CARD FOR      */00292000
                                              /*    "READER".                   */00293000
                          MONTH (12) CHAR (3) INITIAL ('JAN', 'FEB', 'MAR',      00294000
                              'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',          00295000
                              'OCT', 'NOV', 'DEC'),                              00296000
                                              /* MONTH NAMES FOR HEADING.       */00297000
                          RELEASE_NUMBER CHAR (7) VARYING,                       00298000
                                              /* SECOND "EXEC PGM=SGINDEX,PARM="*/00299000
                                              /*    PARAMETER.                  */00300000
                          SYSPRINT_SPACE CHAR (57),                              00301000
                                              /* CONTAINS LAST "EXEC PGM=       */00302000
                                              /*    SGINDEX,PARM=" PARAMETER;   */00303000
                                              /*    REPLACES OPERANDS OF ALL    */00304000
                                              /*    "//SYSPRINT DD" CARDS READ. */00305000
                          DATE_OF_LISTING CHAR (9),                              00306000
                                              /* TODAY'S DATE.                  */00307000
                          SORT_STRING CHAR (44),                                 00308000
                                              /* STRING BEING PASSED TO SORT    */00309000
                                              /*    PROGRAM, COPIED FROM        */00310000
                                              /*    "SORT_RECORD".              */00311000
                          DELIMITER CHAR (1), /* CHARACTER WHICH DELIMITED      */00312000
```

```
/* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

                                    /*    "FIELD".                          */00313000
                    DDNAME CHAR (10) VARYING,                                00314000
                                    /* HOLDS DDNAME OF DD CARD TO BE        */00315000
                                    /*    SEARCHED FOR IN "JCL_SCAN".       */00316000
                    DSNAME CHAR (8),                                         00317000
                                    /* HOLDS SECOND LEVEL OF DSNAME OF      */00318000
                                    /*    DD CARD SCANNED IN                */00319000
                                    /*    "JCL_SCAN".                       */00320000
                    MEMBER CHAR (8),                                         00321000
                                    /* HOLDS MEMBER NAME, IF PRESENT,       */00322000
                                    /*    CORRESPONDING TO DSNAME,          */00323000
                                    /*    ABOVE, IN "JCL_SCAN".             */00324000
                    NON_MACRO_NAME (12) CHAR (8) INIT ('DC', 'AIF', 'AGO',   00325000
                        'ANOP', 'CSECT', 'PRINT', 'END', 'ENTRY', 'DS',      00326000
                        'TITLE', 'SE', 'ME'),                                00327000
                                    /* LIST OF OP CODES WHICH ARE NOT       */00328000
                                    /*    MACRO CODES.                      */00329000
                    SORT_CONTROL CHAR (25)                                   00330000
                        INITIAL (' SORT FIELDS=(1,44,CH,A) '),               00331000
                    RECORD_CONTROL CHAR (27)                                 00332000
                        INITIAL (' RECORD TYPE=F,LENGTH=(44) '),             00333000
                                    /* SORT CONTROL CARD IMAGES.            */00334000
                    CHECK CHAR (2),     /* HOLDS FIRST TWO CARD COLUMNS.    */00335000
                    COPYLIB CHAR (8),   /* LIB RECEIVING MEMBERS IN COPY.   */00336000
                    LNKOLIB CHAR (8))   /* HOLDS NAME OF LIBRARY TO BE      */00337000
                                    /*    LINKED INTO.                      */00338000
                        STATIC;                                              00339000

5     1     ARITHMETIC_VARIABLES:                                            00340000
                DECLARE                                                      00341000
                    (PAGE_COUNT DECIMAL FIXED INITIAL (0),                   00342000
                                    /* PAGE COUNTER.                        */00343000
                    GENERATING_NUMBER DECIMAL FIXED (9) INITIAL (0),         00344000
                                    /* NUMBER GENERATED FOR                 */00345000
                                    /*    SYSIN CARDS.                      */00346000
                    (I, J, K, L) BINARY FIXED,                               00347000
                                    /* WORK VARIABLES.                      */00348000
                    DDNAME_LENGTH BINARY FIXED,                              00349000
                                    /* SAVE AREA FOR LENGTH OF              */00350000
                                    /*    "DDNAME".                         */00351000
                    DICOUNT BINARY FIXED,                                    00352000
                                    /* NUMBER OF SYSUT_ CARDS YET TO        */00353000
                                    /*    BE SCANNED.                       */00354000
                    SKIP_COUNT BINARY FIXED INITIAL (1),                     00355000
                                    /* SKIP FOR NEXT PUT SYSPRINT.          */00356000
                    LINES_PER_PAGE BINARY FIXED INITIAL (60),                00357000
                                    /* LINES PER PAGE, COPIED FROM          */00358000
                                    /*    FIRST EXEC PGM=SGINDEX PARM.      */00359000
                    ENTRY_COUNT FIXED BINARY,                                00360000
                                    /* INDEX ENTRY NUMBER ON THIS PAGE      */00361000
                                    /*    BEING PROCESSED.                  */00362000

/* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

                    LEFT_ENTRY FIXED BINARY,                                 00363000
                                    /* INDEX OF "SORTED_ENTRY" BEING        */00364000
                                    /*    PRINTED.                          */00365000
                    THUMB_SPOT FIXED BINARY,                                 00366000
                                    /* LINE NUMBER ON WHICH THUMB TAB       */00367000
                                    /*    IS TO BE PRINTED.                 */00368000
                    SORT_RETURN_CODE FIXED BINARY (31),                      00369000
                                    /* CODE RETURNED BY SORT;  IGNORED.     */00370000
                    SORT_RECORD_COUNT BINARY FIXED (31) INITIAL (0))         00371000
                                    /* COUNT OF SORT RECORDS OUTPUT.        */00372000
                        STATIC;                                              00373000

6     1     MISCELLANEOUS:                                                   00374000
                DECLARE                                                      00375000
                    1 SORT_RECORD STATIC,                                    00376000
                                    /* RECORD OF INDEX INFORMATION TO       */00377000
                                    /*    BE SORTED.                        */00378000
                        5 ITEM CHAR (8),                                     00379000
                                    /* ACTUAL INDEX ITEM.                   */00380000
                        5 LIB CHAR (8), /* LIBRARY ON WHICH ITEM RESIDES.   */00381000
                        5 USE CHAR (8), /* WAY IN WHICH ITEM IS USED IN     */00382000
                                    /*    SYSGEN.                           */00383000
                        5 STEP CHAR (8),/* STEP IN WHICH REFERENCE OCCURS.  */00384000
                        5 PAGE_NUMBER PICTURE 'ZZZ9',                        00385000
                                    /* ZERO-SUPPRESSED PAGE NUMBER OF       */00386000
                                    /*    LISTING ON WHICH ITEM             */00387000
                                    /*    APPEARS.                          */00388000
                        5 CARD_ID CHAR (8),                                  00389000
                                    /* CARD COLUMNS 73-80 OF CARD ON        */00390000
                                    /*    WHICH ITEM APPEARS.               */00391000

                    SORT_RECORD_AREA DEFINED SORT_RECORD CHAR (44),          00392000
                                    /* 44 CHARACTER STRING CONTAINING       */00393000
                                    /*   ALL INDEX FIELDS.                  */00394000

                    1 CARD_AREA STATIC, /* AREA INTO WHICH CARD IS READ.    */00395000
                        5 CARD CHAR (71),                                    00396000
                                    /* COLUMNS 1-71.                        */00397000
                        5 CONTINUATION_COLUMN CHAR (1) INITIAL (' '),        00398000
                                    /* COLUMN 72.                           */00399000
                        5 CARD_NUMBER PICTURE '99999999',                    00400000
                                    /* COLUMNS 73-80.                       */00401000

                    CARD_IMAGE DEFINED CARD_AREA CHAR (80),                  00402000
                                    /* 80-CHARACTER CARD-IMAGE STRING.      */00403000


                    CARD_NUMBER_AREA DEFINED CARD_AREA POSITION (73)         00404000
                        CHAR (8),   /* CHARACTERS IN 73-80 OF               */00405000
                                    /*    "CARD_IMAGE".                     */00406000
```

```
      /* *                    * S G I N D E X *                    * */00001000

STMT LEVEL NEST

                      NUMBER PICTURE '222,229' STATIC,                          00407000
                                 /* EDIT AREA FOR                             */00408000
                                 /*    "SORT_RECORD_COUNT".                   */00409000

                      1 DATE_SPOT STATIC, /* RECEIVING AREA FOR DATE.          */00410000
                           2 YEAR CHAR (2),                                     00411000
                           2 MONTH_NUMBER CHAR (2),                             00412000
                           2 DAY CHAR (2),                                      00413000

                      DATE_RECEIVING_AREA DEFINED DATE_SPOT CHAR (6),           00414000
                                 /* FULL REDEFINITION.                        */00415000

                      (RECORD_ADDRESS, LENGTH, KEYLOC) ENTRY (CHAR(25), CHAR(27), FIXED BINARY(31),  00416000
                           FIXED BINARY(31), ENTRY, ENTRY),                     00417000
                                 /* SORT MODULE ENTRY ATTRIBUTES.             */00418000
                      INITIAL ENTRY (FIXED BINARY(31)),                         00419000
                                 /* SORT MODULE CONDITION CODE                */00420000
                                 /*    SETTER ATTRIBUTES.                     */00421000

                      (RECORD_GENERATED,  /* EVENT MARKED COMPLETED EACH       */00422000
                                 /*   TIME A COMPLETE INDEX ENTRY             */00423000
                                 /*   IS GENERATED INTO                       */00424000
                                 /*   "SORT_RECORD".                          */00425000

                      INDEX_PRINTED,     /* EVENT MARKED WHEN "SORT_INTER-    */00426000
                                 /*    FACE" TASK COMPLETES.                  */00427000

                      RECORD_ACCEPTED) EVENT STATIC;                            00428000
                                 /* EVENT MARKED COMPLETED WHENEVER           */00429000
                                 /*   "SORT_RECORD" IS COPIED INTO            */00430000
                                 /*   "SORT_STRING".                          */00431000

  7     1          DECLARE                                                      00432000
                      VERIFY ENTRY RETURNS (FIXED BINARY);                      00433000
                                 /* DECLARE VERIFY FOR RELEASE 10.            */00434000
                                 /* NOTE:  DELETE THIS STATEMENT              */00435000
                                 /*    AND PROCEDURE "VERIFY" IF              */00436000
                                 /*    COMPILED UNDER PL/I VERSION            */00437000
                                 /*    2 OR LATER.                            */00438000

      /* *                    * S G I N D E X *                    * */00001000

STMT LEVEL NEST

   8     1          SYSGEN_INDEX_CREATION:                                     /************************************/00443000
                                                                               /* THIS BLOCK INITIALIZES, THEN     */00444000
                                                                               /*    LOOPS FOR EACH JOB STEP,      */00445000
                                                                               /*    INVOKING THE PROPER BLOCK TO  */00446000
                                                                               /*    HANDLE THAT STEP, AND FINALLY */00447000
                                                                               /*    CLEANS UP AND ENDS.           */00448000
                                                                               /************************************/00449000


                           IF LENGTH (PARM) = 0      /* IF PARM IS ABSENT, GO OPEN      */00450000
   9     1                     THEN GO TO OPEN_FILES;                           00451000
                                 /*   FILES.                                    */00452000
  10     1                I, J = INDEX (PARM, ','); /* SEARCH FOR COMMA.          */00453000
  11     1                IF I = 0                  /* IF MISSING, PRETEND IT FOLLOWS. */00454000
  12     1                     THEN I = LENGTH (PARM) + 1;                      00455000
  13     1                IF I ^= 1                 /* UNLESS "PARM=,XXXXXX",     */00456000
  14     1                     THEN STRING = SUBSTR (PARM, 1, I-1);             00457000
                                 /*   PUT FIRST PARM INTO STRING.               */00458000
  15     1                IF J = 0                  /* IF NO OTHER PARMS PRESENT,      */00459000
  16     1                     THEN PARM = '';      /*    THEN EMPTY OUT PARM;  ELSE   */00460000
  17     1                     ELSE PARM = SUBSTR (PARM, I+1);                  00461000
                                 /*   PUT REST OF PARMS INTO "PARM".            */00462000
  18     1                IF I = 1                  /* IF FIRST PARM IS MISSING,       */00463000
  19     1                     THEN GO TO OPEN_FILES;                           00464000
                                 /*   BRANCH.                                   */00465000
  20     1                ON CONVERSION             /* IF FIRST PARM INCLUDES NON-     */00466000
  21     1                     STRING = '60';       /*   NUMERIC, REPLACE WITH DEFAULT.*/00467000
  22     1                LINES_PER_PAGE = STRING;  /* PUT FIRST PARM INTO "LINES_PER_ */00468000
                                 /*   PAGE".                                    */00469000
  23     1                REVERT CONVERSION;        /* RESTORE SYSTEM HANDLING.        */00470000


  24     1          OPEN_FILES:                                                00471000
                        OPEN FILE (SYSIN),         /* OPEN ALL FILES.           */00472000
                             FILE (PRINTER) PAGESIZE (LINES_PER_PAGE) TITLE ('INDEX'),00473000
                             FILE (NEWDECK);                                    00474000
  25     1                ON ENDFILE (SYSIN)        /* AT END OF SYSIN FILE, SET       */00475000
  26     1                     ALL_DONE = '1'B;     /*    "ALL_DONE" SWITCH.     */00476000
  27     1                DATE_RECEIVING_AREA = DATE;                           00477000
                                 /* FETCH TODAY'S DATE.                       */00478000
  28     1                DATE_OF_LISTING = DAY || ' ' || MONTH(MONTH_NUMBER) ||  00479000
                                 ' ' || YEAR;      /* MAKE DATE PRETTY.         */00480000
  29     1                ON ENDPAGE (PRINTER) BEGIN;                           00481000
                                 /* AT END OF PAGE ON SYSPRINT:               */00482000
  31     2                     PAGE_COUNT = PAGE_COUNT + 1;                     00483000
  32     2                     PAGE_NUMBER = PAGE_COUNT;                        00484000
                                 /* BUMP AND EDIT PAGE COUNT.                 */00485000
  33     2                     PUT FILE (PRINTER) EDIT                          00486000
                                    (RELEASE_INFO, DATE_OF_LISTING,           00487000
                                    'SYSGEN INDEX LISTING', LIST_ID, 'PAGE ',  00488000
                                    PAGE_NUMBER)                              00489000
                                    (PAGE, A, X(10), A, COLUMN(50), A, X(10), A,  00490000
```

```
      /* *                    " S O I N D E X "                        * */00001000

STMT LEVEL NEST

                                   COLUMN (110), 2 A);                              00491000
 54     2                  PUT FILE (PRINTER) SKIP
                            ('------ ------ -------') (SKIP (0), COLUMN (50), A);00493000
                                                /* OUTPUT PAGE HEADING.          */00494000
 55     2                  PUT FILE (PRINTER) SKIP (3);                            00495000
 56     2                  END;                                                    00496000
 57     1            STATUS (RECORD_GENERATED) = 0;                                00497000
                                                /* SET STATUS OF EVENT "RECORD_  */00498000
                                                /*    GENERATED" AS 0 TO MEAN    */00499000
                                                /*    INDEX RECORD IS BEING PASSED.*/00500000
 58     1            COMPLETION (RECORD_GENERATED) = '0'B;                         00501000
                                                /* INITIALIZE EVENT VARIABLES TO */00502000
                                                /*  '0'B.                        */00503000
 59     1            COMPLETION (RECORD_ACCEPTED) = '0'B;                          00504000
 60     1            CALL SORT_INTERFACE TASK EVENT (INDEX_PRINTED);               00505000
                                                /* ATTACH "SORT_INTERFACE" TO    */00506000
                                                /*    PASS INDEX ENTRIES TO SORT */00507000
                                                /*    PROGRAM, RECEIVE SORTED    */00508000
                                                /*    ENTRIES BACK, AND PRINT THEM.*/00509000
 61     1            IF LENGTH(PARM) = 0        /* IF NO PARM= WAS INCLUDED ON   */00510000
                                                /*    EXEC PGM=SOINDEX AFTER LINE*/00511000
                                                /*    COUNT DELETED, BRANCH.     */00512000
 62     1                 THEN GO TO SIGNAL_PAGE;                                  00513000
 63     1            I = INDEX (PARM, ','); /* FIND THE END OF THE RELEASE      */00514000
                                                /*    NUMBER (SECOND PARM).      */00515000
 64     1            IF I = 0                   /* IF IT'S THE ONLY PARM THEN:   */00516000
 65     1                 THEN DO;                                                 00517000
 66     1   1                 RELEASE_NUMBER = PARM;                              00518000
 67     1   1                 GO TO INFO_SET;                                     00519000
 68     1   1                 END;               /*   SET AS RELEASE NUMBER WHOLE*/00520000
                                                /*   PARM AND BRANCH.           */00521000
 69     1            RELEASE_NUMBER = SUBSTR (PARM, 1, I-1);                       00522000

 70     1       INFO_SET:                                                         00523000
                  RELEASE_INFO = 'RELEASE ' || RELEASE_NUMBER;                    00524000
                                                /* SET UP RELEASE INFO FOR HEADER.*/00525000
 71     1            IF LENGTH (RELEASE_NUMBER) > 2                               00526000
 72     1                 THEN RELEASE_NUMBER =                                   00527000
                                 SUBSTR (RELEASE_NUMBER, 1, 2);                   00528000
                                                /* LIMIT RELEASE NUMBER FOR      */00529000
                                                /*    SEQUENCE NUMBERING TO TWO  */00530000
                                                /*    DIGITS.                    */00531000
 73     1            ON CONVERSION              /* IF RELEASE_NUMBER IS NOT NUMERIC*/00532000
                                                /*   REPLACE WITH ZEROS.        */00533000
 74     1                 RELEASE_NUMBER = '00';                                  00534000
 75     1            GENERATING_NUMBER = RELEASE_NUMBER;                           00535000
                                                /* USE RELEASE NUMBER AS LOW-ORDER */00536000
                                                /*    DIGITS OF CARD NUMBERS.    */00537000
 76     1            REVERT CONVERSION;         /* REVERT TO SYSTEM PROCESSING.  */00538000
 77     1            IF I = 0                   /* IF NO MORE PARMS, BRANCH.     */00539000
 78     1                 THEN GO TO SIGNAL_PAGE;                                 00540000
```

```
      /* *                    " S O I N D E X "                        * */00001000

STMT LEVEL NEST

 79     1            STRING = SUBSTR (PARM, I+1);                                  00541000
                                                /* GET REST OF PARM.             */00542000
 80     1            I = INDEX (STRING, ','); /* FIND END OF PARM THREE.       */00543000
 81     1            IF I = 0                   /* IF NO TRAILING COMMA, FAKE ONE. */00544000
 82     1                 THEN I = LENGTH (STRING) + 1;                           00545000
 83     1            OBJECT_CARD_DELETION_REQUESTED =                              00546000
                         SUBSTR (STRING, 1, I-1) = 'OBJDEL';                      00547000
                                                /* SET SWITCH AS THE PRESENCE OF */00548000
                                                /*    "OBJDEL" DICTATES.         */00549000
 84     1            IF I >= LENGTH (STRING)    /* IF NOTHING FOLLOWS THIRD PARM,*/00550000
                                                /*    BRANCH.                    */00551000
 85     1                 THEN GO TO SIGNAL_PAGE;                                 00552000
 86     1            STRING = SUBSTR (STRING, I+1);                                00553000
                                                /* DELETE THE THIRD PARM.        */00554000
 87     1            I = INDEX (STRING, ','); /* FIND THE NEXT COMMA.          */00555000
 88     1            IF I = 0                   /* IF MISSING, JIMMY UP A COMMA. */00556000
 89     1                 THEN I = LENGTH (STRING) + 1;                           00557000
 90     1            SEQUENCE_SUPPRESS =                                           00558000
                         SUBSTR (STRING, 1, I-1) = 'NOSEQ';                       00559000
                                                /* SET "SEQUENCE_SUPPRESS" AS    */00560000
                                                /*    PRESENCE OF "NOSEQ" DICTATES.*/00561000
 91     1            IF I >= LENGTH (STRING)    /* IF THIS IS LAST PARM, BRANCH. */00562000
 92     1                 THEN GO TO SIGNAL_PAGE;                                 00563000
 93     1            STRING = SUBSTR (STRING, I+1);                                00564000
                                                /* DELETE PREVIOUS PARM.         */00565000
 94     1            I = INDEX (STRING, ','); /* FIND COMMA.                   */00566000
 95     1            IF I = 0                   /* IF MISSING, FIX.              */00567000
 96     1                 THEN I = LENGTH (STRING) + 1;                           00568000
 97     1            LIST_ID = SUBSTR (STRING, 1, I-1);                            00569000
                                                /* GET LAST IDENTIFICATION.      */00570000
 98     1            IF I > LENGTH (STRING)     /* IF THAT'S ALL, BRANCH.        */00571000
 99     1                 THEN GO TO SIGNAL_PAGE;                                 00572000
100     1            SYSPRINT_SPACE = SUBSTR (STRING, I+1);                        00573000
                                                /* ANYTHING LEFT IS A REPLACEMENT*/00574000
                                                /*    FOR "//SYSPRINT DD" OPERANDS.*/00575000
101     1            MODIFY_SYSPRINT_REQUESTED = '1'B;                             00576000
                                                /* NOTE THAT SYSPRINT CARDS ARE  */00577000
                                                /*   TO BE MODIFIED.             */00578000

102     1       SIGNAL_PAGE:                                                      00579000
                  SIGNAL ENDPAGE (PRINTER); /* PRINT FIRST PAGE HEADER.       */00580000
103     1            GO TO FLUSH;               /* ADVANCE TO FIRST STEP.        */00581000

                                                                                 00582000

                                                /***** END OF HOUSEKEEPING. ********/00583000
```

```
        /* *                    " S O I N D E X "                    * */00001000

STMT LEVEL NEST

 84     1        EXEC_CARD_PROCESS:                                          00584000
                    USR = 'STEPNAME';          /* SET "USR" AS STEPNAME.    */00585000
                    LIB = '        ';          /* CLEAR LIB IN SORT RECORD. */00586000
 85     1                  I = INDEX (CARD, ' EXEC ');                       00587000
                                               /* FIND COLUMN IN WHICH ' EXEC '*/00588000
                                               /*    STARTS.                */00589000
 87     1                  J = I - 3;          /* SUBTRACTING 3 GIVES LENGTH OF*/00590000
                                               /*    STEPNAME.              */00591000
 88     1                  ITEM = SUBSTR (CARD, 3, J);                       00592000
                                               /* PUT STEPNAME INTO ITEM.   */00593000
 89     1                  STEP = ITEM;        /* ALSO COPY INTO STEP NAME. */00594000
 90     1                  CALL OUTPUT_SORT_KEY; /* OUTPUT THAT RECORD; RETURN HERE.*/00595000
 91     1                  USE = 'EXEC    ';   /* SET USE AS EXEC.          */00596000
 92     1                  STRING = SUBSTR (CARD, 10);                       00597000
                                               /* STRING BEGINS WITH "EXEC ...".*/00598000
 93     1                  SPACE_BYPASS = '1'B; /* SET TO BYPASS "EXEC" AND SPACES.*/00599000
 94     1                  EQUAL_SCAN_REQUEST = '1'B; /* SET TO BYPASS "PGM=".*/00600000
 95     1                  CALL FIELD_SCAN;    /* SCAN FOR PROGRAM NAME, RETURNED*/00601000
                                               /*    IN "FIELD".  RETURN HERE.*/00602000
 96     1                  IF ERROR_OCCURRED                                 00603000
 97     1                       THEN GO TO FLUSH; /* ADVANCE TO NEXT "EXEC" ON ERROR.*/00604000
 98     1                  ITEM = FIELD;       /* PICK UP FIELD AS ITEM OF NEXT*/00605000
                                               /*    INDEX ENTRY.           */00606000
 99     1                  CALL OUTPUT_SORT_KEY; /* OUTPUT ENTRY;  RETURN HERE.*/00607000
100     1                  IF FIELD = 'IEWL'   /* BRANCH TO HANDLER FOR      */00608000
101     1                       THEN CALL LINK_PROCESS;                      00609000
                                               /*    PROGRAM BEING EXECUTED. */00610000
102     1                       ELSE IF FIELD = 'ASMBLR'                     00611000
103     1                            THEN CALL ASMBLR_PROCESS;               00612000
104     1                            ELSE IF FIELD = 'IEBCOPY'               00613000
105     1                                 THEN CALL COPY_PROCESS;            00614000
106     1                                 ELSE GO TO FLUSH;                  00615000
107     1                  IF EXEC_CARD         /* IF LAST PROCESS READ NEXT EXEC*/00616000
                                               /*    CARD, THEN GO PROCESS IT.*/00617000
108     1                       THEN GO TO EXEC_CARD_PROCESS;                00618000

109     1        FLUSH:                                                      00619000
                    READ_TO_EXEC = '1'B;       /* NOTE TO "READER" WE WANT NEXT*/00620000
                                               /*    EXEC CARD.             */00621000
110     1                  CALL READER;        /* READ SYSIN.               */00622000
111     1                  IF ¬ ALL_DONE       /* LOOP TO HANDLE STEPS TILL */00623000
112     1                       THEN GO TO EXEC_CARD_PROCESS;                00624000
                                               /*    SYSIN EXHAUSTED.       */00625000
113     1                  NUMBER = SORT_RECORD_COUNT;                       00626000
114     1                  PUT FILE (PRINTER) SKIP (3)                       00627000
                               EDIT (NUMBER, ' INDEX ENTRIES COLLECTED')     00628000
                               (COLUMN(5), F 4);  /* PRINT COUNT.          */00629000
115     1                  CLOSE FILE (SYSIN),  /* CLOSE ALL FILES BUT "PRINTER".*/00630000
                               FILE (READDECK); /*    THIS WILL FREE SOME CORE.*/00631000
116     1                  STATUS (RECORD_GENERATED) = 4;                    00632000
                                               /* INDICATE TO "SORT_INTERFACE"*/00633000

        /* *                    " S O I N D E X "                    * */00001000

STMT LEVEL NEST

117     1                  COMPLETION (RECORD_GENERATED) = '1'B;             00634000
                                               /*    THAT ALL ENTRIES HAVE BEEN*/00635000
                                               /*    PASSED.                */00636000
118     1                  WAIT (INDEX_PRINTED); /* WAIT FOR "SORT_INTERFACE" TO*/00637000
                                               /*    COMPLETE.              */00638000
119     1                  CLOSE FILE (PRINTER); /* CLOSE PRINTER FILE.       */00639000
120     1                  RETURN;              /* END OF PROGRAM.           */00640000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

141     1       LINK_PROCESS: PROCEDURE;        /*****************************************/00041000
                                                /* THIS BLOCK EXTRACTS THE SYSLMOD */00042000
                                                /*    LIBRARY NAME, OUTPUTS INDEX  */00043000
                                                /*    ENTRIES FOR HOST LINKAGE     */00044000
                                                /*    EDITOR CONTROL CARDS, AND    */00045000
                                                /*    EXITS AT THE END OF THE      */00046000
                                                /*    CONTROL CARD FILE.           */00047000
                                                /*****************************************/00048000

144     2               LMODLIB = '        ';   /* CLEAR LMODLIB TO SPACES.        */00049000
143     2               READ_TO_EXEC = '0'B;    /* TURN OFF READING TO EXEC CARD.  */00050000
146     2               DDNAME = 'SYSLMOD ';    /* SET DDNAME FOR "JCL_SCAN" TO    */00051000
                                                /*    FIND THE SYSLMOD DD CARD.    */00052000
145     2               CALL JCL_SCAN;          /* GO GET SYSLMOD DSNAME AND       */00053000
                                                /*    MEMBER, IF PRESENT.          */00054000
148     2               IF ALL_DONE             /* END OF FILE IS AN ERROR;  EXIT. */00055000
147     2                   THEN GO TO LINK_EXIT;                                    00056000
128     2               LMODLIB = DSNAME;       /* SAVE DSNAME AS LMODLIB.         */00057000
149     2               IF MEMBER = ' '         /* IF NO MEMBER WAS LISTED, GO ON  */00058000
                                                /*    TO CONTROL CARDS.            */00059000
150     2                   THEN IF JCL                                              00060000
151     2                            THEN GO TO CONTROL_SCAN;                        00061000
152     2                            ELSE GO TO SAT_CONTROL_LOOP;                    00062000
153     2               ITEM = MEMBER;          /* ELSE SET MEMBER AS ITEM.        */00063000
134     2               USE = 'LINK    ';       /* SET 'LINK' AS USE.              */00064000
135     2               LIB = LMODLIB;          /* LMODLIB IS THE LIB.             */00065000
136     2               CALL OUTPUT_SORT_KEY;   /* OUTPUT ENTRY.                   */00066000

137     2       CONTROL_SCAN:                                                        00067000
                                                /* "READER" IS TO READ TO NEXT     */00068000
                        READ_TO_NON_JCL = '1'B; /*    NON-JCL CARD.                */00069000
138     2               CALL READER;            /* GO GET FIRST CONTROL CARD.      */00070000
139     2               IF ALL_DONE             /* END OF FILE SHOULD NOT OCCUR.   */00071000
140     2                   THEN GO TO LINK_EXIT;                                    00072000

141     2       SAT_CONTROL_LOOP:                                                    00073000
                        READ_TO_NON_JCL = '0'B; /* ALL OTHER "READER" CALLS SHOULD */00074000
142     2               READ_TO_NON_CONTIN = '1'B;/*    GET NEXT NON-CONTINUATION    */00075000
                                                /*    CARD.                        */00076000

143     2       CONTROL_LOOP:                                                        00077000
                        STRING = CARD;          /* PUT CARD INTO EXAMINATION       */00078000
                                                /*    STRING.                      */00079000
144     2               SPACE_BYPASS = '1'B;    /* SET UP FOR "FIELD_SCAN" TO      */00080000
145     2               SPECIAL_SCAN_REQUEST = '0'B;/*    GET OPERATION FIELD.      */00081000
146     2               CALL FIELD_SCAN;        /* GET OPERATION FIELD.            */00082000
147     2               IF ERROR_OCCURRED       /* IGNORE ERRORS (MESSAGE HAS      */00083000
                                                /*    ALREADY BEEN OUTPUT).        */00084000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

                            THEN GO TO NEXT_CONTROL;                             00085000
149     2               USE = FIELD;            /* PUT OP CODE INTO USE.           */00086000
                                                /* ALIAS AND ENTRY CARDS BRANCH.   */00087000
150     2               IF USE = 'ALIAS' | USE = 'ENTRY'                             00088000
151     2                   THEN GO TO SET_LIB;                                      00089000
                                                /* INCLUDE AND CHANGE CARDS        */00090000
154     2               IF USE = 'INCLUDE' | USE = 'CHANGE'                          00091000
153     2                   THEN GO TO SCAN_LIB;                                     00092000
                                                /*    BRANCH HERE.                 */00093000
156     2               IF USE = 'REPLACE'      /* REPLACE CARDS BRANCH HERE.      */00094000
155     2                   THEN GO TO CLEAR_LIB;                                    00095000
158     2               IF USE ^= 'NAME'        /* NAME CARDS FALL THROUGH, ALL    */00096000
157     2                   THEN GO TO NEXT_CONTROL;                                 00097000
                                                /*    ELSE BRANCHES.               */00098000
158     2               USE = 'LINK    ';       /* NAME CARDS GENERATE 'LINK' USE. */00099000

159     2       SET_LIB:                                                             00100000
                        LIB = LMODLIB;          /* GET LMODLIB AS LIB.             */00101000
160     2               GO TO GET_ITEMS;        /* GO GET OPERANDS.                */00102000

161     2       CLEAR_LIB:                                                           00103000
                        LIB = '        ';       /* CLEAR LIB FIELD.                */00104000
162     2               GO TO GET_ITEMS;        /* GO GET OPERANDS.                */00105000

163     2       SCAN_LIB:                                                            00106000
                        CALL FIELD_SCAN;        /* GET FIRST OPERAND ON INCLUDE    */00107000
                                                /*    AND CHANGE CARDS.            */00108000
164     2               IF ERROR_OCCURRED       /* IF FIELD NOT FOUND, IGNORE.     */00109000
165     2                   THEN GO TO NEXT_CONTROL;                                 00110000
166     2               LIB = FIELD;            /* USE FIELD AS LIB.               */00111000

167     2       GET_ITEMS:                                                           00112000
                        SPACE_BYPASS = '0'B;    /* NOTE THAT "FIELD_SCAN" CALLS    */00113000
                                                /*    SHOULD NOT BYPASS ANYTHING.  */00114000
168     2               CALL OUTPUT_ALL_KEYS;   /* MAKE AN OUTPUT ENTRY FOR EACH   */00115000
                                                /*    OPERAND.                     */00116000
169     2               IF ERROR_OCCURRED       /* NEXT CARD ON ERRORS.            */00116100
170     2                   THEN GO TO NEXT_CONTROL;                                 00116200
171     2               IF USE = 'INCLUDE' | USE = 'CHANGE'                          00116300
                            THEN                /* ON INCLUDE AND CHANGE CARDS,    */00116400
172     2                       IF SUBSTR (STRING, 2, 1) = ','                       00116500
173     2                           THEN GO TO SCAN_LIB;                             00116600
                                                /*    IF MORE ENTRIES ARE PRESENT, */00116700
                                                /*    GO SET UP NEXT ONE.          */00116800

174     2       NEXT_CONTROL:                                                        00117000
                        CALL HEADER;            /* HEAD FOR NEXT CONTROL CARD.     */00118000
```

```
       /* *                    * S O I N D E X *                    * */00001000

SINI LEVEL NEST

175     2               AT ALL_DONE             /* EXIT ON EOF.                 */00719000
176     2                    THEN GO TO LINK_EXIT;                               00720000
177     2               CHECK = SUBSTR (CARD, 1, 2);                             00721000
178     2               IF CHECK = '/*' | CHECK = '//'                           00722000
179     2                    THEN GO TO LINK_EXIT;                               00723000
                                            /* LOOP UNTIL CONTROL CARD FILE    */00724000
                                            /*    ENDS.                        */00725000
180     2               GO TO CONTROL_LOOP;  /* LOOP.                        */00726000

181     2          LINK_EXIT:                                                    00727000
                   END LINK_PROCESS;        /*************************************/00728000
```

```
       /* *                    * S O I N D E X *                    * */00001000

SINI LEVEL NEST

182     1          ASMBLR_PROCESS:  PROCEDURE;      /*************************************/00729000
                                                    /* THIS BLOCK EXTRACTS THE SYSPUNCH*/00730000
                                                    /*    DSNAME AND MEMBER TO OUTPUT  */00731000
                                                    /*    AS AN INDEX ENTRY, LOOPS TO  */00732000
                                                    /*    OUTPUT EACH MACRO INSTRUC-   */00733000
                                                    /*    TION AS AN INDEX ENTRY, AND  */00734000
                                                    /*    EXITS AT END OF CONTROL CARD */00735000
                                                    /*    FILE.                        */00736000
                                                    /*************************************/00737000
183     2               READ_TO_EXEC = '0'B;         /* TURN OF HEADING FOR EXEC CARD. */00738000
184     2               DSNAME = 'SYSPUNCH ';        /* "JCL_SCAN" IS TO FIND SYSPUNCH */00739000
185     2               CALL JCL_SCAN;               /*    DD CARD;  RETURN HERE.      */00740000
186     2               IF ALL_DONE                  /* EOF IS ERROR HERE; EXIT.       */00741000
187     2                    THEN GO TO ASMBLR_EXIT;                                 00742000
188     2               IF MEMBER = ' '              /* IF NOT FOUND, SKIP IT.         */00743000
189     2                    THEN IF JCL                                             00744000
190     2                         THEN GO TO MACRO_SCAN;                             00745000
191     2                         ELSE GO TO SET_MACRO_LOOP;                         00746000
192     2               USE = 'SYSPUNCH';            /* SET SYSPUNCH AS USE.           */00747000
193     2               LIB = DSNAME;                /* LIB IS DSNAME.                 */00748000
194     2               ITEM = MEMBER;               /* ITEM IS MEMBER.                */00749000
195     2               CALL OUTPUT_SOME_KEY;        /* OUTPUT ENTRY.                  */00750000

196     2          MACRO_SCAN:                                                   00751000
                                                    /* INVOKE "HEADER" TO SCAN DOWN TO */00752000
                   READ_TO_NON_JCL = '1'B;          /*    FIRST NON-JCL CARD.         */00753000
197     2               CALL HEADER;                                             00754000
198     2               IF ALL_DONE                  /* EOF IS ERROR HERE;  EXIT.      */00755000
199     2                    THEN GO TO ASMBLR_EXIT;                             00756000

200     2          SET_MACRO_LOOP:                                               00757000
                   READ_TO_NON_JCL = '0'B;          /* OTHER READS SHOULD GO TO NEXT  */00758000
201     2               READ_TO_NON_CONTIN = '1'B;  /*    NON-CONTINUATION CARD.      */00759000
202     2               SPACE_BYPASS = '1'B;         /* ALL "FIELD_SCAN" CALLS SHOULD  */00760000
203     2               EQUAL_SCAN_REQUEST = '0'B;  /*    SPACE-BYPASS, BUT NOT SCAN  */00761000
                                                    /*    FOR EQUAL SIGNS.            */00762000
204     2               USE = 'MACRO   ';            /* ALL ENTRIES HERE WILL HAVE USE */00763000
205     2               LIB = '        ';            /*    AS "MACRO", LIB BLANK.      */00764000

206     2          MACRO_LOOP:                                                   00765000
                   IF SUBSTR (CARD, 1, 1) = '*'                                 00766000
207     2                    THEN GO TO NEXT_MACRO;                              00767000
                                                    /* IF THIS IS A COMMENT CARD, SKIP */00768000
                                                    /*    IT.                         */00769000
208     2               STRING = CARD;               /* SET CARD AS EXAMINATION STRING. */00770000
209     2               CALL FIELD_SCAN;             /* SCAN FOR OP CODE.              */00771000
210     2               IF ERROR_OCCURRED            /* SKIP ANY BAD CARDS.            */00772000
```

```
/* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

211     2                   THEN GO TO NEXT_MACRO;                          00773000
212     2               DO  I = 1 TO 12;          /* CHECK ALL NON-MACRO N/MES.  */00774000
213     2   1               IF FIELD = NON_MACRO_NAME(I)                      00775000
214     2   1                   THEN GO TO NEXT_MACRO;                        00776000
                                                  /* IF THIS OP CODE IS IN THE NON- */00777000
                                                  /*    MACRO LIST, SKIP IT.    */00778000
215     2   1               END;                                              00779000
                                                  /* IF OP CODE IS NOT A MACRO,  */00780000
                                                  /*    IGNORE IT.              */00781000
216     2               ITEM = FIELD;             /* INDEX ITEM IS MACRO.       */00782000
217     2               CALL OUTPUT_SORT_KEY;     /* OUTPUT ENTRY.              */00783000


218     2       NEXT_MACRO:                                                   00784000
                    CALL HEADER;                  /* GET NEXT NON-CONTINUATION CARD. */00785000
219     2           IF ALL_DONE                   /* EOF IS ERROR HERE;  EXIT.  */00786000
220     2               THEN GO TO ASSEMB_EXIT;                               00787000
221     2           CHECK = SUBSTR (CARD, 1, 4);                              00788000
222     2           IF CHECK = '/**' | CHECK = '///'                          00789000
223     2               THEN GO TO ASSEMB_EXIT;                               00790000
                                                  /* IF END OF CONTROL CARD FILE,  */00791000
                                                  /*    EXIT.                   */00792000
224     2           GO TO MACRO_LOOP;             /* LOOP BACK TO PROCESS NEXT CARD. */00793000


225     2       ASSEMB_EXIT:                                                  00794000
                    END ASSEMB_PROCESS;           /*************************/00795000
```

```
/* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

226     1   COPY_PROCESS:  PROCEDURE;             /*************************/00796000
                                                  /* THIS BLOCK SCANS THE DSNAMES OF */00797000
                                                  /*    THE SYSUT1 AND SYSUT2 CARDS, */00798000
                                                  /*    OUTPUTS INDEX ENTRIES FOR    */00799000
                                                  /*    EACH MEMBER COPIED, CLEANS   */00800000
                                                  /*    UP, AND EXITS.               */00801000
                                                  /*************************/00802000


227     2               UTCOUNT = 2;              /* WE ARE GOING TO FIND 2 SYSUT   */00803000
                                                  /*    DD CARDS.                   */00804000
228     2               DSNAME = 'SYSUT';         /* WE'LL PICK THEM UP IN EITHER   */00805000
                                                  /*    ORDER.                      */00806000
229     2               READ_TO_EXEC = '0'B;      /* TURN OFF READING TO EXEC CARD. */00807000


230     2       JCL_CHECK:                                                    00808000
                    CALL JCL_SCAN;                /* GO GET A SYSUT CARD.       */00809000
231     2           IF ALL_DONE                   /* EOF HERE IS AN ERROR;  EXIT. */00810000
232     2               THEN GO TO COPY_EXIT;                                 00811000
233     2           IF DSNAME = ' '               /* IF NOT FOUND, GO PROCESS THE */00812000
                                                  /*    CONTROL CARDS.          */00813000
234     2               THEN IF JCL                                           00814000
235     2                   THEN GO TO COPY_CONTROL_CARDS;                    00815000
236     2                   ELSE GO TO SET_COPY_LOOP;                         00816000
237     2           FIELD = DSNAME;               /* COPY DSNAME INTO FIELD.    */00817000
238     2           IF SUBSTR (CARD, 8, 1) = '1'                              00818000
239     2               THEN CALL SYSUT1_SET;                                 00819000
                                                  /* THE SYSUT1 DSNAME GETS OUTPUT  */00820000
                                                  /*    IN AN INDEX ENTRY BY        */00821000
                                                  /*    SYSUT1_SET;  THE SYSUT2     */00822000
240     2               ELSE COPYLIB = DSNAME;                                00823000
                                                  /*    DSNAME IS SAVED AS THE      */00824000
                                                  /*    COPY (RECEIVING) LIB.       */00825000
241     2           UTCOUNT = UTCOUNT - 1;        /* ONE LESS TO DO.            */00826000
242     2           IF UTCOUNT ^= 0               /* IF WE HAVEN'T DONE BOTH, LOOP. */00827000
243     2               THEN GO TO JCL_CHECK;                                 00828000


244     2       COPY_CONTROL_CARDS:                                           00829000
                                                  /* ASK "HEADER" TO ADVANCE TO     */00830000
                    READ_TO_NON_JCL = '1'B;       /*    FIRST COPY CONTROL CARD.    */00831000
245     2           CALL HEADER;                                              00832000
246     2           IF ALL_DONE                   /* AS USUAL, EOF IS ERROR.    */00833000
247     2               THEN GO TO COPY_EXIT;                                 00834000


248     2       SET_COPY_LOOP:                                                00835000
                    READ_TO_NON_JCL = '0'B;       /* ALL OTHER "HEADER" CALLS IN THIS*/00836000
249     2               READ_TO_NON_CONTIN = '1'B;/*    BLOCK READ TO NEXT NON-     */00837000
                                                  /*    CONTINUATION CARD.          */00838000
```

```
      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

250     2         COPY_LOOP:                                                  00839000
                      STRING = CARD;          /* SET CARD AS EXAMINATION STRING. */00840000
251     2             SPACE_BYPASS = '1'B;    /* CALL TO "FIELD_SCAN" SHOULD   */00841000
252     2             EQUAL_SCAN_REQUEST = '0'B;/*    BYPASS SPACES, THEN PICK UP */00842000
253     2             CALL FIELD_SCAN;        /*    OPERATION FIELD.           */00843000
254     2             IF ERROR_OCCURRED       /* SKIP ERROR CARDS.             */00844000
255     2                  THEN GO TO NEXT_CONTROL_CARD;                        00845000
256     2             IF FIELD = 'SELECT' | FIELD = 'S'                         00846000
257     2                  THEN GO TO SELECT_HANDLE;                           00847000
                                            /* BRANCH TO HANDLE SELECT CARDS. */00848000
258     2             IF FIELD = 'EXCLUDE' | FIELD = 'E'                        00849000
259     2                  THEN GO TO EXCLUDE_HANDLE;                          00850000
                                            /* BRANCH TO HANDLE EXCLUDE CARDS. */00851000
260     2             IF FIELD = 'REMBER'     /* BRANCH TO HANDLE OLD REMBER   */00852000
                                            /*   CARDS.                      */00853000
261     2                  THEN GO TO OLD_MEMBER_CARDS;                        00854000
262     2             IF FIELD ^= 'COPY'      /* FALL THROUGH FOR COPY CARDS,  */00855000
                                            /*    IGNORE ALL OTHERS.         */00856000
263     2                  THEN GO TO NEXT_CONTROL_CARD;                       00857000
264     2             I = INDEX (STRING, 'OUTDD=');                            00858000
                                            /* FIND OUTDD OPERAND.           */00859000
265     2             IF I = 0                /* IF MISSING, SKIP THE WHOLE MESS.*/00860000
266     2                  THEN GO TO NEXT_CONTROL_CARD;                       00861000
267     2             STRING = SUBSTR (STRING, I+6);                           00862000
                                            /* ADVANCE TO OPERAND.           */00863000
268     2             SPACE_BYPASS = '0'B;    /* SCAN FOR OPERAND FIELD, WITH  */00864000
269     2             CALL FIELD_SCAN;        /*    NO SPACE BYPASSING.        */00865000
270     2             IF ERROR_OCCURRED       /* IF ALL SORTCMED UP, FORGIT IT. */00866000
271     2                  THEN GO TO NEXT_CONTROL_CARD;                       00867000
272     2             COPYLIB = FIELD;        /* SAVE COPYLIB DDNAME AS RECEIVING*/00868000
                                            /*    LIBRARY.                   */00869000
273     2             I = INDEX (CARD, 'INDD=');/* GO SEARCH CARD FOR INDD.    */00870000
274     2             IF I = 0                /* IF MISSING, IGNORE.           */00871000
275     2                  THEN GO TO NEXT_CONTROL_CARD;                       00872000
276     2             STRING = SUBSTR (CARD, I+5);                             00873000
                                            /* START STRING WITH OPERAND.    */00874000
277     2             CALL FIELD_SCAN;        /* EXTRACT OPERAND.              */00875000
278     2             IF ERROR_OCCURRED       /* SKIP OUT ON ERROR.            */00876000
279     2                  THEN GO TO NEXT_CONTROL_CARD;                       00877000
280     2             CALL SYSUT1_SET;        /* GO OUTPUT COPYFROM ITEM.      */00878000


281     2         OLD_MEMBER_CARDS:                                           00879000
                      USE = 'COPYTO  ';       /* "MEMBER" CARDS ARE            */00880000
                                              /*   PRE-RELEASE-19 VERSION OF   */00881000
                                              /*   "SELECT" CARDS;  HANDLE SIMU-*00882000
                                              /*   LARLY.                      */00883000
282     2             I = INDEX (STRING, 'NAME=');                             00884000
                                              /* THESE USE "NAME=" INSTEAD OF  */00885000
                                              /*    "MEMBER=".                 */00886000
283     2             IF I = 0                /* SKIP BAD CARDS.               */00887000
284     2                  THEN GO TO NEXT_CONTROL_CARD;                       00888000
```

```
      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

285     2                 I = I + 5;          /* ADVANCE PAST KEYWORD.         */00889000
286     2                 GO TO GET_MEMBERS;  /* GO PROCESS OPERANDS.          */00890000


287     2         SELECT_HANDLE:                                              00891000
                      USE = 'COPYTO  ';       /* SET "COPYTO" AS USE.          */00892000
288     2             SELECT_MEMBER_SCAN = '1'B; /* TELL "OUTPUT_ALL_KEYS" TO BYPASS*/00893000
                                              /*    "...,R)," OPERANDS.        */00894000
289     2             GO TO MEMBER_SCAN;      /* GO SCAN MEMBERS.              */00895000


290     2         EXCLUDE_HANDLE:                                             00896000
                      USE = 'EXCLUDE';        /* SET "EXCLUDE" AS USE.         */00897000


291     2         MEMBER_SCAN:                                                00898000
                      I = INDEX (STRING, 'MEMBER=');                           00899000
                                              /* FIND MEMBER= OPERAND.         */00900000
294     2             IF I ^= 0               /* IF  "MEMBER=" FOUND:          */00901000
293     2                  THEN DO;                                            00902000
294     2  1                    I = I + 7;    /*    ADVANCE PAST KEYWORD;      */00903000
295     2  1                 GO TO GET_MEMBERS;                               00904000
296     2  1                 END;             /*    GO PROCESS OPERANDS.       */00905000
297     2             I = INDEX (STRING, 'M='); /* PROCESS ABBREVIATION SIMULARLY.*/00906000
298     2             IF I = 0                /* IF THAT'S MISSING, SKIP CARD. */00907000
299     2                  THEN GO TO NEXT_CONTROL_CARD;                       00908000
300     2             I = I + 2;                                               00909000


301     2         GET_MEMBERS:                                                00910000
                      STRING = SUBSTR (STRING, I);                             00911000
                                              /* ADVANCE TO OPERAND AREA.      */00912000
302     2             SPACE_BYPASS = '0'B;    /* NOTE THAT SCAN FOR FIELDS IS  */00913000
                                              /*    NOT NECESSARY.             */00914000
303     2             LIB = COPYLIB;          /* SET OUTDD LIB AS LIB.         */00915000
304     2             CALL OUTPUT_ALL_KEYS;   /* GO OUTPUT A SORT ENTRY FOR    */00916000
                                              /*    EACH MEMBER PRESENT.       */00917000
305     2             SELECT_MEMBER_SCAN = 0B; /* RESET SWITCH.                */00918000


306     2         NEXT_CONTROL_CARD:                                          00919000
                      CALL HEADER;            /* READ NEXT NON-CONTINUATION CARD.*/00920000
307     2             IF ALL_DONE             /* EXIT ON EOF (ERROR).          */00921000
308     2                  THEN GO TO COPY_EXIT;                               00922000
309     2             CHECK = SUBSTR (CARD, 1, 2);                             00923000
310     2             IF CHECK = '/*' | CHECK = '//'                           00924000
311     2                  THEN GO TO COPY_EXIT;                               00925000
                                              /* IF CONTROL CARD FILE ENDED,   */00926000
                                              /*    EXIT.                      */00927000
312     2             GO TO COPY_LOOP;        /* ELSE, LOOP BACK.              */00928000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

313     2       COPY_EXIT:                                                  00929000
                    SELECT_MEMBER_SCAN = '0'B;/* INSURE SWITCH IS OFF.     */00930000
314     2           END COPY_PROCESS;       /*****************************/00931000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

315     1       SYSUT1_SET:  PROCEDURE;     /*****************************/00932000
                                            /* THIS SHORT BLOCK OUTPUTS AN */00933000
                                            /*    INDEX ENTRY FOR USE     */00934000
                                            /*    "COPYFROM" WHETHER ITEM */00935000
                                            /*    CAME FROM SYSUT1 DD CARD OR */00936000
                                            /*    INDD= IN COPY CONTROL CARD. */00937000
                                            /*****************************/00938000

316     2               USE = 'COPYFROM';   /* USE IS "COPYFROM".         */00939000
317     2               ITEM = FIELD;       /* ITEM COMES FROM FIELD;  IT'S THE*/00940000
                                            /*    NAME OF THE LIBRARY COPIED */00941000
                                            /*    FROM IN EITHER CASE.    */00942000
318     2               LIB = '        ';   /* LIB FIELD IS BLANK.        */00943000
319     2               CALL OUTPUT_SORT_KEY; /* OUTPUT INDEX ENTRY.      */00944000
320     2           END SYSUT1_SET;         /*****************************/00945000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

321     1       OUTPUT_ALL_KEYS:  PROCEDURE;/*****************************/00946000
                                            /* THIS BLOCK GETS FIELDS FROM THE */00947000
                                            /*    STRING AND OUTPUTS INDEX */00948000
                                            /*    ENTRIES FOR THOSE ITEMS, */00949000
                                            /*    LOOPING AS LONG AS THE   */00950000
                                            /*    FIELDS ARE DELIMITED WITH */00951000
                                            /*    COMMAS.                  */00952000
                                            /*****************************/00953000


322     2       NEXT_KEY:                                                   00954000
                    CALL FIELD_SCAN;        /* GO GET NEXT FIELD.         */00955000
323     2           IF ERROR_OCCURRED       /* GET OUT ON ANY ERROR.      */00956000
324     2               THEN GO TO ALL_OUT_EXIT;                            00957000
325     2           IF SELECT_MEMBER_SCAN   /* IF WE'RE PROCESSING "SELECT" */00958000
                                            /*    CONTROL CARD OF "IEBCOPY", */00959000
326     2               THEN DO;            /*    THE FOLLOWING IS REQUIRED: */00960000
327     2   1               IF FIELD = 'M' & DELIMITER = ')'                 00961000
                                            /*    IF THE OPERAND IS "... M)", */00962000
328     2   1                   THEN DO;                                     00963000
329     2   2                       STRING = SUBSTR (STRING, 2);             00964000
                                            /*    THEN DELETE THE ")",     */00965000
330     2   2                       DELIMITER = SUBSTR (STRING, 1, 1);       00966000
                                            /*    PICK THE NEXT DELIMITER, */00967000
331     2   2                       GO TO COMMA_CHECK;                       00968000
                                            /*    AND BYPASS OUTPUTING FIELD. */00969000
332     2   2                       END;                                     00970000
333     2   1                   END;                                         00971000
334     2           ITEM = FIELD;           /* SET THIS FIELD AS ITEM.    */00972000
335     2           CALL OUTPUT_SORT_KEY;   /* OUTPUT INDEX ENTRY.        */00973000


336     2       COMMA_CHECK:                                                00974000
                    IF DELIMITER ^= ','     /* IF DELIMITER IS NOT A COMMA, */00975000
                                            /*    SKEDADDLE.               */00976000
337     2               THEN GO TO ALL_OUT_EXIT;                            00977000
338     2           IF SUBSTR (STRING, 2, 1) ^= ' '                         00978000
339     2               THEN GO TO NEXT_KEY;                                00979000
                                            /* IF COMMA IS NOT FOLLOWED BY */00980000
                                            /*    BLANK, LOOP FOR NEXT KEY. */00981000
340     2           IF CONTINUATION_COLUMN = ' '                            00982000
341     2               THEN GO TO ALL_OUT_EXIT;                            00983000
                                            /* IF CARD IS NOT CONTINUED, ERROR.*/00984000
                                            /*    IN THAT CASE, EXIT.      */00985000
342     2           READ_TO_NON_CONTIN = '0'B;/* ASK "READER" TO GIVE US NEXT */00986000
343     2           CALL READER;            /*    CARD, WHATEVER IT IS.    */00987000
344     2           READ_TO_NON_CONTIN = '1'B;/* RESET FOR OTHER'S USE.     */00988000
345     2           I = VERIFY (CARD, ' '); /* FIND FIRST NON-BLANK IN CARD. */00989000
346     2           IF I = 0                /* BLANK CARD IS A NO-NO; EXIT. */00990000
347     2               THEN GO TO ALL_OUT_EXIT;                            00991000
348     2           STRING = SUBSTR (CARD, I);                              00992000
                                            /* PICK UP CONTINUED OPERAND. */00993000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

349     2               GO TO NEXT_KEY;     /* GO PROCESS IT.             */00994000


350     2       ALL_OUT_EXIT:                                               00995000
                    END OUTPUT_ALL_KEYS;    /*****************************/00996000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

351    1        OUTPUT_SORT_KEY:  PROCEDURE;      /************************************/00997000
                                                  /* THIS SMORT BLOCK PASSES THE      */00998000
                                                  /*    INDEX ENTRY TO "SORT_INTER-   */00999000
                                                  /*    FACE" AND KEEPS COUNT.        */01000000
                                                  /************************************/01001000


352    2              COMPLETION (RECORD_GENERATED) = '1'B;                  01002000
                                                  /* LET "SORT_INTERFACE" KNOW RECORD*/01003000
                                                  /*    IS READY.                     */01004000
353    2              WAIT (RECORD_ACCEPTED);     /* WAIT FOR IT TO MAKE A COPY.     */01005000
354    2              COMPLETION (RECORD_ACCEPTED) = '0'B;                   01006000
                                                  /* RESET EVENT FOR NEXT TIME       */01007000
                                                  /*    AROUND.                      */01008000
355    2              SORT_RECORD_COUNT = SORT_RECORD_COUNT + 1;             01009000
                                                  /* BUMP COUNT OF SORT RECORDS.     */01010000
356    2              END OUTPUT_SORT_KEY;         /************************************/01011000


        /* *                    " S G I N D E X "                    * */00001000

START LEVEL NEST

357    1        READER:  PROCEDURE;               /************************************/01012000
                                                  /* THIS BLOCK READS SYSGEN STAGE II*/01013000
                                                  /*    RECORDS, INSERTING SEQUENCE  */01014000
                                                  /*    NUMBERS AND REPLACING        */01015000
                                                  /*    SYSPRINT CARDS IF REQUESTED, */01016000
                                                  /*    AND ECHOING THEM ON THE      */01017000
                                                  /*    LISTING.  THIS IS REPEATED   */01018000
                                                  /*    UNTIL A CARD OF THE TYPE     */01019000
                                                  /*    REQUESTED IS READ.  CONTROL  */01020000
                                                  /*    IS THEN RETURNED WITH THE    */01021000
                                                  /*    CARD'S COLUMNS 1-71 IN       */01022000
                                                  /*    "CARD", COLUMN 72 IN "CON-   */01023000
                                                  /*    TINUATION_COLUMN",           */01024000
                                                  /*    "EXEC_CARD" SET TO 1B IF     */01025000
                                                  /*    AN EXEC CARD WAS READ, AND   */01026000
                                                  /*    'JCL' SET TO '1'B IF A "//"  */01027000
                                                  /*    CARD WAS READ.               */01028000
                                                  /************************************/01029000

358    2        READ_ANOTHER:                                                  01030000
                    EXEC_CARD = '0'B;             /* RESET "EXEC_CARD" AND "JCL"     */01031000
359    2              JCL = '0'B;                 /*    BITS.                        */01032000
360    2              PREV_CONTIN = CONTINUATION_COLUMN;                     01033000
                                                  /* SAVE CONTINUATION COLUMN DURING */01034000
                                                  /*    READ.                        */01035000
361    2              GET FILE (SYSIN)            /* READ IN CARD IMAGE.             */01036000
                        EDIT (CARD_IMAGE) (A (80));                         01037000
362    2              IF ALL_DONE                 /* EXIT ON END OF INPUT FILE.      */01038000
363    2                 THEN GO TO READ_EXIT;                                01039000
364    2              IF OBJECT_CARD_DELETION_REQUESTED                      01040000
                          & SUBSTR (CARD, 1, 1) = ' '                       01041000
365    2                 THEN GO TO READ_ANOTHER;                           01042000
                                                  /* SKIP OBJECT CARDS, IF REQUESTED.*/01043000
366    2              IF ^ SEQUENCE_SUPPRESS      /* UNLESS SEQUENCING HAS BEEN      */01044000
367    2                 THEN DO;                 /*    SUPPRESSED BY REQUEST:       */01045000
368    2  1                 GENERATING_NUMBER = GENERATING_NUMBER + 10000;     01046000
                                                  /* INCREMENT CARD NUMBER BY 10000. */01047000
369    2  1                 CARD_NUMBER = GENERATING_NUMBER;                   01048000
                                                  /* PUT INTO CARD.                  */01049000
370    2  1                 END;                                               01050000
371    2              IF SUBSTR (CARD, 1, 2) ^= '//'                         01051000
372    2                 THEN GO TO ECHO;         /* NON-JCL BYPASS.                 */01052000
373    2              JCL = '1'B;                 /* NOTE THAT JCL CARD WAS FOUND.   */01053000
374    2              IF MODIFY_SYSPRINT_REQUESTED                           01054000
                          & SUBSTR (CARD, 1, 11) = '//SYSPRINT '            01055000
375    2                 THEN DO;                 /* FOR SYSPRINT CARDS:             */01056000
376    2  1                 CARD = '//SYSPRINT DD ' || SYSPRINT_SPACE;         01057000
                                                  /*    REPLACE WITH SUPPLIED SPACE  */01058000
                                                  /*    ALLOCATION.                  */01059000
377    2  1                 GO TO ECHO;           /*    SKIP DOWN TO ECHO.           */01060000
378    2  1                 END;                                               01061000
```

```
        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

379     2                 I = INDEX (CARD, ' ');    /* FIND FIRST BLANK.                */01062000
380     2                 IF I = 0                  /* NO BLANKS?  SURELY YOU JEST.     */01063000
381     2                      THEN GO TO ECHO;     /*     (ALSO SURELY YOU BRANCH.)    */01064000
382     2                 STRING = SUBSTR (CARD, I);/* CHOP OFF NAME FIELD.             */01065000
383     2                 I = VERIFY (STRING, ' '); /* SCAN FOR FIRST NON-BLANK.        */01066000
384     2                 IF I = 0                  /* MORE INFINITELY NONSENSE; SKIP.  */01067000
385     2                      THEN GO TO ECHO;                                           01068000
386     2                 IF SUBSTR (STRING, I, 5) = 'EXEC '                              01069000
387     2                      THEN DO;             /* FOR EXEC CARDS:                  */01070000
                                                    /*    NOTE THAT EXEC CARD FOUND.    */01071000
388     2   1                      EXEC_CARD = '1'B;                                      01071500
389     2   1                      SKIP_COUNT = 3;/*    TRIPLE SPACE LISTING BEFORE   */01072000
390     2   1                      GO TO ECHO;    /*    PRINTING IT.                  */01073000
391     2   1                      END;                                                   01074000
392     2                 IF SUBSTR (STRING, I, 4) = 'JOB '                               01075000
393     2                      THEN SKIP_COUNT = 3;                                       01076000
                                                    /* TRIPLE SPACE BEFORE JOB CARDS.  */01077000


394     2           ECHO:                                                               01078000
                    PUT FILE (PRINTER)              /* OUTPUT ECHO OF CARD.            */01079000
                         SKIP (SKIP_COUNT)                                              01080000
                         EDIT (CARD_IMAGE) (COLUMN(26), A);                             01081000
395     2                 SKIP_COUNT = 1;           /* RESET TO SINGLE SPACE.          */01082000
396     2                 PUT FILE (HENDECK)        /* ALSO OUTPUT A COPY ON HENDECK.  */01083000
                              EDIT (CARD_IMAGE) (A);                                    01084000
397     2                 IF READ_TO_EXEC           /* IF EXECUTE CARD REQUESTED AND   */01085000
                              THEN                  /*    FOUND, EXIT; IF REQUESTED BUT*/01086000
398     2                      IF EXEC_CARD         /*    NOT FOUND, LOOP.             */01087000
                                   THEN                                                 01088000
399     2                                GO TO READ_EXIT;                                01089000
400     2                           ELSE                                                 01090000
400     2                                GO TO READ_ANOTHER;                             01091000
401     2                 IF READ_TO_NON_JCL        /* IF NON-JCL REQUESTED AND FOUND, */01092000
                              THEN                  /*    EXIT;  IF REQUESTED BUT NOT  */01093000
402     2                      IF JCL               /*    FOUND, LOOP.                 */01094000
                                   THEN                                                 01095000
403     2                                GO TO READ_ANOTHER;                             01096000
404     2                           ELSE                                                 01097000
404     2                                GO TO READ_EXIT;                                01098000
405     2                 IF READ_TO_NON_CONTIN     /* IF NON-CONTINUATION REQUESTED   */01099000
                              THEN                  /*    AND FOUND, EXIT;  IF RE-     */01100000
406     2                      IF PREV_CONTIN = ' '                                      01101000
                                   THEN             /*    QUESTED BUT NOT FOUND, LOOP. */01102000
407     2                                GO TO READ_EXIT;                                01103000
408     2                           ELSE                                                 01104000
                                    GO TO READ_ANOTHER;                                 01105000

409     2           READ_EXIT:                                                          01106000
                         CARD_ID = CARD_NUMBER_AREA;                                    01107000
                                                    /* INSERT INTO SORT_RECORD THE     */01108000


        /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

                                                    /*    CARD ID NUMBER OF CARD BEING */01109000
                                                    /*    RETURNED FOR PROCESSING.     */01110000
                                                    /*********************************/01111000
410     2                 END READER;
```

```
/P *                    " S C I N D E X "                    * */00001000

STMT LEVEL NEST

411    1        FIELD_SCAN:  PROCEDURE;          /*************************************/01112000
                                                 /* THIS BLOCK SCANS FOR A FIELD      */01113000
                                                 /*    WITHIN THE CHARACTER STRING    */01114000
                                                 /*    "STRING".  IF "SPACE_BYPASS"   */01115000
                                                 /*    IS SET, NON-BLANKS BEFORE THE  */01116000
                                                 /*    FIRST BLANK AND THEN BLANKS    */01117000
                                                 /*    UP TO THE NEXT NON-BLANK ARE   */01118000
                                                 /*    IGNORED.  IF "EQUAL_SCAN_RE-   */01119000
                                                 /*    QUEST" IS SET, MORE CHAR-      */01120000
                                                 /*    ACTERS ARE DISCARDED UNTIL     */01121000
                                                 /*    AN EQUAL SIGN IS FOUND.        */01122000
                                                 /*    THE FIELD RETURNED IS THE      */01123000
                                                 /*    STRING BEGINNING AT THE PLACE  */01124000
                                                 /*    INDICATED ABOVE, AND TERMINA-  */01125000
                                                 /*    TING WITH A COMMA, SPACE,      */01126000
                                                 /*    OR OPEN OR CLOSE PARENTHESIS.  */01127000
                                                 /*    (IF THE STRING BEGINS WIT-     */01128000
                                                 /*    ANY OF THESE DELIMITERS,       */01129000
                                                 /*    THEY ARE DISCARDED.)  THE      */01130000
                                                 /*    RESULTING STRING IS RETURNED   */01131000
                                                 /*    IN "FIELD".  THE DELIMITING    */01132000
                                                 /*    CHARACTER IS RETURNED IN       */01133000
                                                 /*    "DELIMITER".  CHARACTERS       */01134000
                                                 /*    PROCESSED ARE REMOVED FROM     */01135000
                                                 /*    "STRING" UP TO BUT EXCLUDING   */01136000
                                                 /*    THE DELIMITER.                 */01137000
                                                 /*************************************/01138000

412    2                 ERROR_OCCURRED = '0'B;   /* RESET ERROR INDICATOR.            */01139000
413    2                 IF ¬ SPACE_BYPASS        /* IF SPACE BYPASS NOT REQUESTED,    */01140000
414    2                      THEN GO TO EQUAL_CHECK;                                    01141000
                                                 /*    BYPASS THE BYPASSING PROCESS.  */01142000
415    2                 I = INDEX (STRING, ' '); /* FIND THE FIRST BLANK.             */01143000
416    2                 IF I = 0                 /* IF NO BLANKS, ERROR.              */01144000
417    2                      THEN GO TO ERROR;                                          01145000
418    2                 IF I ¬= 1                                                       01146000
419    2                      THEN STRING = SUBSTR (STRING, I);                          01147000
                                                 /* DELETE LEADING CHARACTERS IF      */01148000
                                                 /*    ANY.                           */01149000
420    2                 I = VERIFY (STRING, ' '); /* FIND NEXT NON-BLANK.             */01150000
421    2                 IF I = 0                 /* IF THE REST IS BLANK, ERROR.      */01151000
422    2                      THEN GO TO ERROR;                                          01152000
423    2                 STRING = SUBSTR (STRING, I);                                    01153000
                                                 /* DELETE LEADING BLANKS.            */01154000

424    2           EQUAL_CHECK:                                                         01155000
                       IF ¬ EQUAL_SCAN_REQUEST   /* IF NOT REQUESTED, SKIP.           */01156000
425    2                   THEN GO TO FIND_FIELD;                                      01157000
426    2                  I = INDEX (STRING, '=') + 1;                                 01158000
                                                 /* FIND THE EQUAL SIGN.              */01159000
```

```
/* *                    " S C I N D E X "                    * */00001000

STMT LEVEL NEST

427    2                 IF I = 1                 /* NO EQUAL SIGN IS AN ERROR.        */01160000
428    2                      THEN GO TO ERROR;                                          01161000
429    2                 STRING = SUBSTR (STRING, I);                                    01162000
                                                 /* CLEAR OUT UP TO THE EQUAL SIGN.   */01163000

430    2           FIND_FIELD:                                                          01164000
                       I = VERIFY (STRING, ' ,()');                                    01165000
431    2                  IF I ¬= 0                /* DELETE ANY LEADING DELIMITERS.    */01166000
432    2                      THEN STRING = SUBSTR (STRING, I);                          01167000
433    2                  SCAN_STRING = STRING || ' ,()';                              01168000
                                                 /* IN FOLLOWING SCANS, GIVE HIGHER   */01169000
                                                 /*    VALUES TO FAILING SCANS.       */01170000
434    2                  I = INDEX (SCAN_STRING, ' ');                                01171000
435    2                  J = INDEX (SCAN_STRING, ',');                                01172000
436    2                  K = INDEX (SCAN_STRING, '(');                                01173000
437    2                  L = INDEX (SCAN_STRING, ')');                                01174000
                                                 /* FIND FIRST OCCURRANCE OF EACH     */01175000
                                                 /*    DELIMITER.                     */01176000
438    2                  I = MIN (I, J, K, L) - 1;                                    01177000
                                                 /* FIND THE FIRST ONE; LESS ONE      */01178000
                                                 /*    IS FIELD LENGTH.               */01179000
439    2                  IF I > LENGTH (STRING)   /* IF ALL WE FOUND IS THOSE TRAIL-   */01180000
440    2                       THEN GO TO ERROR;   /*    ING ONES WE ADDED, ERROR.      */01181000
441    2                  FIELD = SUBSTR (STRING, 1, I);                               01182000
                                                 /* EXTRACT THE RESULT.               */01183000
442    2                  STRING = SUBSTR (STRING, I+1);                               01184000
                                                 /* DELETE PROCESSED PART OF STRING.  */01185000
443    2                  DELIMITER = SUBSTR (STRING, 1, 1);                           01186000
                                                 /* RETURN DELIMITER SEPARATELY.      */01187000
444    2                  GO TO EXIT;              /* DAH-A-DA-DAH-A-DA-A-DAT'S ALL,    */01188000
                                                 /*    FOLKS.                         */01189000

445    2           ERROR:                                                               01190000
                       PUT FILE (PRINTER)        /* OUTPUT CRYPTIC ERROR MESSAGE.     */01191000
                           EDIT ('****** ERROR IN ABOVE CARD') (A);                    01192000
446    2                  SKIP_COUNT = 3;          /* TRIPLE SPACE BEFORE NEXT CARD.    */01193000
447    2                  ERROR_OCCURRED = '1'B;   /* SET ERROR INDICATOR.              */01194000

448    2           EXIT:                                                                01195000
                       END FIELD_SCAN;           /*************************************/01196000
```

```
      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

449    1       JCL_SCAN:  PROCEDURE;                 /*****************************************/01197000
                                                     /* THIS BLOCK WILL SEARCH THROUGH        */01198000
                                                     /*    DD CARDS UNTIL THE CARD            */01199000
                                                     /*    WHOSE NAME (OR PARTIAL NAME)       */01200000
                                                     /*    IS STORED IN "DDNAME" IS           */01201000
                                                     /*    FOUND.  THE DSNAME= PARAMETER      */01202000
                                                     /*    IS THEN FOUND AND THE SECOND       */01203000
                                                     /*    LEVEL OF THE DATA SET NAME         */01204000
                                                     /*    IS EXTRACTED AND RETURNED          */01205000
                                                     /*    IN "DSNAME".  IF A MEMBER          */01206000
                                                     /*    NAME IS PRESENT, IT IS             */01207000
                                                     /*    RETURNED IN "MEMBER".              */01208000
                                                     /*****************************************/01209000

450    2                DDNAME_LENGTH = LENGTH (DDNAME);                                        01210000
                                                     /* SAVE LENGTH OF DDNAME TO BE        */01211000
                                                     /*    SCANNED.                        */01212000
451    2                READ_TO_EXEC = '0'B;         /* READS SHOULD ADVANCE TO NEXT       */01213000
452    2                READ_TO_NON_JCL = '0'B;      /*    NON-CONTINUATION CARD.          */01214000
453    2                READ_TO_NON_CONTIN = '1'B;                                           01215000
454    2                DSNAME = '        ';         /* CLEAR DSNAME AND MEMBER.           */01216000
455    2                MEMBER = '        ';                                                 01217000

456    2        DDNAME_CHECK:                                                                01218000
                    CALL READER;                     /* GET ANOTHER CARD.                  */01219000
457    2            IF ^ JCL                         /* IF WE RUN OUT OF JCL CARDS,        */01220000
                                                     /*    EXIT, DADDY.                    */01221000
458    2                THEN GO TO JCL_EXIT;                                                 01222000
459    2            IF SUBSTR (CARD, 3, DDNAME_LENGTH) ^= DDNAME                             01223000
460    2                THEN GO TO DDNAME_CHECK;                                             01224000
                                                     /* IF THIS AIN'T IT, LOOP.            */01225000

461    2        DSNAME_SCAN:                                                                 01226000
                    I = INDEX (CARD, 'DSNAME=');                                           01227000
                                                     /* FIND DSNAME= PARAMETER.            */01228000
462    2            IF I ^= 0                        /* IF WE FOUND IT, BRANCH.            */01229000
463    2                THEN DO;                                                             01230000
464    2  1                 I = I + 3;                                                      01231000
465    2  1                 GO TO FIND_PERIOD;                                              01232000
466    2  1                 END;                                                            01233000
467    2            I = INDEX (CARD, 'DSN='); /* ALSO GIVE DSN= A CHANCE.                 */01234000
468    2            IF I ^= 0                                                              01235000
469    2                THEN GO TO FIND_PERIOD;                                            01236000
470    2            IF CONTINUATION_COLUMN = ' '                                           01237000
471    2                THEN GO TO JCL_EXIT;                                               01238000
                                                     /* IF WE'RE NOT CONTINUED, EXIT.      */01239000
472    2            READ_TO_NON_CONTIN = '0'B;/* READ THE NEXT CARD IMAGE.                */01240000
473    2            READ_TO_NON_JCL = '0'B;                                                01241000
474    2            CALL READER;                                                           01242000
```

```
      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

475    2            IF ALL_DONE                      /* EOF HERE IS ERROR;  EXIT.          */01243000
476    2                THEN GO TO JCL_EXIT;                                               01244000
477    2            GO TO DSNAME_SCAN;               /* LOOP AND TRY AGAIN.                */01245000

478    2        FIND_PERIOD:                                                                 01246000
                    STRING = SUBSTR (CARD, I+4);                                           01247000
                                                     /* ADVANCE PAST DSN= OR DSNAME=.      */01248000
479    2            I = INDEX (STRING, '.');                                               01249000
                                                     /* FIND THE FIRST PERIOD.             */01250000
480    2            IF I = 0 | I > 8                                                       01251000
481    2                THEN GO TO EXTRACT_DSNAME;                                         01252000
                                                     /* IF NO SECOND-LEVEL NAME, GET       */01253000
                                                     /*    FIRST-LEVEL NAME.               */01254000
482    2            STRING = SUBSTR (STRING, I+1);                                         01255000
                                                     /* ADVANCE PAST FIRST LEVEL NAME      */01256000
                                                     /*    AND PERIOD.                     */01257000

483    2        EXTRACT_DSNAME:                                                              01258000
                    SPACE_BYPASS = '0'B;             /* FIELD_SCAN FOR DSNAME WITHOUT      */01259000
484    2            EQUAL_SCAN_REQUEST = '0'B;/*    SPACE-BYPASS OR EQUAL-SEARCH.       */01260000
485    2            CALL FIELD_SCAN;                                                       01261000
486    2            IF ERROR_OCCURRED                /* EXIT ON ERROR.                     */01262000
487    2                THEN GO TO JCL_EXIT;                                               01263000
488    2            DSNAME = FIELD;                  /* STORE RESULT.                      */01264000
489    2            IF DELIMITER ^= '('              /* IF MEMBER NAME IS NOT PRESENT,     */01265000
490    2                THEN GO TO JCL_EXIT;/*    GET THE ____ OUTA THERE.              */01266000
491    2            CALL FIELD_SCAN;                 /* SCAN FOR MEMBER NAME.              */01267000
492    2            IF ERROR_OCCURRED                /* EXIT ON ERROR.                     */01268000
493    2                THEN GO TO JCL_EXIT;                                               01269000
494    2            MEMBER = FIELD;                  /* STORE RESULT.                      */01270000

495    2        JCL_EXIT:                                                                    01271000
                    END JCL_SCAN;                    /*****************************************/01272000
```

```
      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

496     1        SORT_INTERFACE: PROCEDURE;       /************************************/01273000
                                                  /*  THIS PROCEDURE IS ATTACHED AS   */01274000
                                                  /*    A SUBTASK.  IT INVOKES THE    */01275000
                                                  /*    SORT ROUTINE WHICH ENTERS     */01276000
                                                  /*    "INDEX_ENTRY_SENDER" TO FETCH */01277000
                                                  /*    EACH OF THE SORT RECORDS      */01278000
                                                  /*    COLLECTED BY THE MOTHER TASK. */01279000
                                                  /*    SORT THEN ENTERS "INDEX_      */01280000
                                                  /*    PRINTER" TO PRINT EACH ENTRY. */01281000
                                                  /*    WHEN SORT RETURNS WE FINISH   */01282000
                                                  /*    OFF THE PRINTOUT AND EXIT.    */01283000
                                                  /************************************/01284000


497     2              DECLARE                                                    01285000
                            SORTED_ENTRY (*) CHAR (44) CONTROLLED,               01286000
                                                  /* TABLE TO HOLD FIRST COLUMN OF */01287000
                                                  /*    INDEX ENTRIES OF EACH PAGE.*/01288000
                            RIGHT_RECORD LIKE SORT_RECORD,                       01289000
                                                  /* STRUCTURE TO BREAK UP ENTRY FOR*/01290000
                                                  /*    RIGHT COLUMN OF PAGE.     */01291000
                            RIGHT_RECORD_AREA DEFINED RIGHT_RECORD CHAR (44);    01292000
                                                  /* STRING OF FULL STRUCTURE.     */01293000


498     2              CALL INKSRT3 (SORT_CONTROL, RECORD_CONTROL, 102400,        01294000
                            SORT_RETURN_CODE, INDEX_ENTRY_SENDER, INDEX_PRINTER);01295000
                                                  /* CALL SORT, WHICH WILL USE     */01296000
                                                  /*    "INDEX_ENTRY_SENDER" TO GET*/01297000
                                                  /*    RECORDS, "INDEX_PRINTER" TO*/01298000
                                                  /*    PRINT THEM.               */01299000
499     2              IF ENTRY_COUNT <= LINES_PER_PAGE                           01300000
                                                  /* WE MUST NOW FLUSH OUT THE LAST*/01301000
                                                  /*    PAGE.  ARE WE WORKING ON THE*/01302000
500     2                   THEN DO;               /*    FIRST COLUMN OF THE PAGE? */01303000
                                                  /* IF SO:                        */01304000
501     2   1                   LEFT_ENTRY = 0;   /*    WE WILL START PRINTING WITH*/01305000
                                                  /*    THE FIRST ELEMENT OF "SORTED_*/01306000
                                                  /*    ENTRY".                   */01307000
502     2   1                   RIGHT_RECORD_AREA = SORTED_ENTRY (ENTRY_COUNT);    01308000
                                                  /*    SAVE LAST ENTRY FOR THUMB */01309000
                                                  /*    TAB.                      */01310000
503     2   1                   END;                                               01311000
504     2                   ELSE ENTRY_COUNT = LINES_PER_PAGE;                     01312000
                                                  /* ELSE, SET LOOP TO STOP AT PAGE*/01313000
                                                  /*    END.                      */01314000
505     2              DO  I = LEFT_ENTRY+1 TO ENTRY_COUNT;                       01315000
                                                  /* LOOP FOR REMAINING ENTRIES.   */01316000
506     2   1               SORT_RECORD_AREA = SORTED_ENTRY (I);                   01317000
                                                  /* PUT ENTRY INTO STRUCTURE.     */01318000
507     2   1               PUT FILE (PRINTER)   /* PRINT THE ENTRY.              */01319000
                                EDIT (SORT_RECORD)                               01320000


      /* *                    " S G I N D E X "                    * */00001000

STMT LEVEL NEST

                                (SKIP (SKIP_COUNT), 3 A(10), A(9), A(6), A(8));  01321000
508     2   1               SKIP_COUNT = 1;       /* INSURE SINGLE SPACE.      */01322000
509     2   1               END;                                                   01323000
510     2              CALL PRINT_THUMB_TAB;      /* GO PRINT LAST THUMB TABS. */01324000
511     2              FREE SORTED_ENTRY;         /* FREE UP ARRAY.            */01325000
512     2              RETURN;                    /* SUB-TASK IS DONE.         */01326000
```

```
        /* *                    " S O R T N D E X "                    * */00001000

STMT LEVEL NEST

513    2        INDEX_ENTRY_SENDER:  PROCEDURE CHAR (44);                          01327000
                                     /************************************/01328000
                                     /*  THIS BLOCK IS ENTERED AS SORT   */01329000
                                     /*    EXIT POINT 15.  IT WAITS FOR  */01330000
                                     /*    THE MOTHER TASK TO COMPLETE   */01331000
                                     /*    THE NEXT RECORD.  WHEN THE    */01332000
                                     /*    RECORD IS COPIED OUT THE      */01333000
                                     /*    MOTHER IS SIGNALED TO PROCESS */01334000
                                     /*    ANOTHER AND WE RETURN THE     */01335000
                                     /*    RECORD TO SORT/MERGE.  WHEN   */01336000
                                     /*    ALL RECORDS HAVE BEEN PASSED, */01337000
                                     /*    PREPARATIONS ARE MADE TO      */01338000
                                     /*    PRINT AND THE FINAL RETURN    */01339000
                                     /*    IS MADE.                      */01340000
                                     /************************************/01341000

514    3                WAIT (RECORD_GENERATED);  /* WAIT FOR MOTHER TO GENERATE A  */01342000
                                                  /*    SORT ENTRY.                 */01343000
515    3                COMPLETION (RECORD_GENERATED) = '0'B;                       01344000
                                                  /* RESET FOR NEXT GO 'ROUND.      */01345000
516    3                IF STATUS (RECORD_GENERATED) ^= 0                           01346000
517    3                     THEN GO TO ALL_TRANSMITTED;                            01347000
                                                  /* IF ALL ENTRIES HAVE BEEN TRANS-*/01348000
                                                  /*    MITTED, BRANCH.             */01349000
518    3                SORT_STRING = SORT_RECORD_AREA;                             01350000
                                                  /* COPY OUT NEW RECORD.           */01351000
519    3                COMPLETION (RECORD_ACCEPTED) = '1'B;                        01352000
                                                  /* LET MOTHER GO TO WORK ON THE   */01353000
                                                  /*    NEXT ONE.                   */01354000
520    3                CALL IHETSAC (12);        /* SET RETURN TO SHOW RECORD      */01355000
                                                  /*    IS COMING.                  */01356000
521    3                RETURN (SORT_STRING);     /* RETURN.STRING TO BE SORTED.    */01357000

522    3        ALL_TRANSMITTED:                                                    01358000
                     ENTRY_COUNT = -1;            /* SET UP FOR PRINTED OUTPUT.  SET*/01359000
                                                  /*    ENTRY_COUNT TO INDICATE     */01360000
                                                  /*    PAGE EJECTION REQUIRED.     */01361000
523    3                THUMB_SPOT = LINES_PER_PAGE - 1;                            01362000
                                                  /* RESERVE LAST 2 LINES FOR THUMB */01363000
                                                  /*    TABS.                       */01364000
524    3                LINES_PER_PAGE = LINES_PER_PAGE - 9;                        01365000
                                                  /* DECREASE "LINES_PER_PAGE" BY   */01366000
                                                  /*    COUNT OF HEADERS, BLANK     */01367000
                                                  /*    LINES, ETC.                 */01368000
525    3                CALL IHETSAC (8);         /* SET RETURN CODE TO SHOW ALL    */01369000
                                                  /*    RECORDS PASSED.             */01370000
526    3                RETURN;                   /* RETURN.                        */01371000
527    3                END INDEX_ENTRY_SENDER;   /************************************/01372000
```

```
        /* *                    " S O R T N D E X "                    * */00001000

STMT LEVEL NEST

528    2        INDEX_PRINTER:  PROCEDURE (SORT_STRING_RETURNED);                   01373000
                                     /************************************/01374000
                                     /* THIS BLOCK IS CALLED BY THE SORT*/01375000
                                     /*    ROUTINE AS EXIT POINT J5.     */01376000
                                     /*    IT PRINTS THE PAGE IN A TWO-  */01377000
                                     /*    COLUMN FORMAT.  LEFT-HAND     */01378000
                                     /*    ENTRIES ARE SAVED UNTIL THE   */01379000
                                     /*    CORRESPONDING RIGHT-HAND      */01380000
                                     /*    ENTRIES ARE RECEIVED, AT      */01381000
                                     /*    WHICH TIME BOTH ARE PRINTED.  */01382000
                                     /************************************/01383000


529    3                DECLARE                                                     01384000
                            SORT_STRING_RETURNED CHAR (44);                      01385000
                                                  /* DEFINE PARAMETER.              */01386000


530    3                IF ENTRY_COUNT < 0        /* IF THIS IS THE FIRST TIME      */01387000
531    3                     THEN DO;             /*    THROUGH:                    */01388000
532    3   1                      CALL IHETSAC (4);                                 01389000
                                                  /* SET RETURN CODE FOR NORMAL.    */01390000
533    3   1                      ALLOCATE SORTED_ENTRY (LINES_PER_PAGE);           01391000
                                                  /* ALLOCATE CORE FOR ARRAY WITH   */01392000
                                                  /*    ENOUGH ROOM TO HOLD LEFT    */01393000
                                                  /*    HALF OF PAGE.               */01394000
534    3   1                      GO TO DO_HEADER;                                  01395000
535    3   1                      END;              /* GO PRINT PAGE HEADER.        */01396000
536    3                ENTRY_COUNT = ENTRY_COUNT + 1;                              01397000
                                                  /* ADVANCE TO NEXT ENTRY.         */01398000
537    3                IF ENTRY_COUNT > LINES_PER_PAGE                             01399000
538    3                     THEN GO TO PRINT_NEXT_LINE;                            01400000
                                                  /* IF WE'RE ON THE SECOND COLUMN, */01401000
                                                  /*    BRANCH.                     */01402000
539    3                SORTED_ENTRY (ENTRY_COUNT) = SORT_STRING_RETURNED;          01403000
                                                  /* SAVE ENTRY IN STRING.          */01404000
540    3                GO TO PRINT_EXIT;         /* THAT'S ALL WE DO, IN THAT CASE.*/01405000

541    3        PRINT_NEXT_LINE:                                                    01406000
                     LEFT_ENTRY = ENTRY_COUNT - LINES_PER_PAGE;                     01407000
                                                  /* CALCULATE ENTRY NUMBER FOR     */01408000
                                                  /*    LEFT ENTRY TO BE PRINTED.   */01409000
542    3                IF LEFT_ENTRY > LINES_PER_PAGE                              01410000
                                                  /* IF WE'VE FILLED THE PAGE,      */01411000
543    3                     THEN GO TO THUMB_TAB;                                  01412000
                                                  /*    THEN BRANCH.                */01413000
544    3                SORT_RECORD_AREA = SORTED_ENTRY (LEFT_ENTRY);               01414000
545    3                RIGHT_RECORD_AREA = SORT_STRING_RETURNED;                   01415000
                                                  /* PUT LEFT AND RIGHT ENTRIES INTO*/01416000
                                                  /*    STRUCTURES FOR PRINTING.    */01417000
546    3                PUT FILE (PRINTER)        /* PRINT THE LINE.                */02418000
```

```
START LEVEL TEST

                        EDIT (SORT_RECORD, RIGHT_RECORD)                        01419000
                        (SKIP(SKIP_COUNT), X (3 A(10), A(9), A(6), A(8), A(14)));01420000
547     1           SKIP_COUNT = 1;                                             01421000
548     1           GO TO PRINT_EXIT;       /* GO BACK FOR NEXT ONE.           */01422000

549     1       THUMB_TAB;                                                      01423000
                    CALL PRINT_THUMB_TAB;   /* PUT THUMB TABS ON PAGE BOTTOM.  */01424000

550     1       DO_HEADER;                                                      01425000
                    PUT FILE (PRINTER)      /* EJECT AND PRINT COLUMN HEADERS. */01426000
                        EDIT (' ITEM    LINENAME    USE    STEP    PAGE    CARD')01427000
                          (ITEM    LINENAME    USE    STEP    PAGE    CARD')    01428000
                          (LINE(1), A);     /*    NOTE THAT THIS RAISES THE    */01429000
                                            /*    "ENDPAGE" CONDITION.         */01430000
551     1           ENTRY_COUNT = 1;        /* RESET FOR NEXT PAGE.            */01431000
552     1           SORTED_ENTRY (1) = SORT_STRING_RETURNED;                    01432000
                                            /* STASH FIRST LINE.               */01433000
553     1           SKIP_COUNT = 2;         /* DOUBLE SPACE FIRST LINE.        */01434000

554     1       PRINT_EXIT;                                                     01435000
                    RETURN;                                                     01436000
555     1           END INDEX_PRINTER;      /*********************************/01437000
```

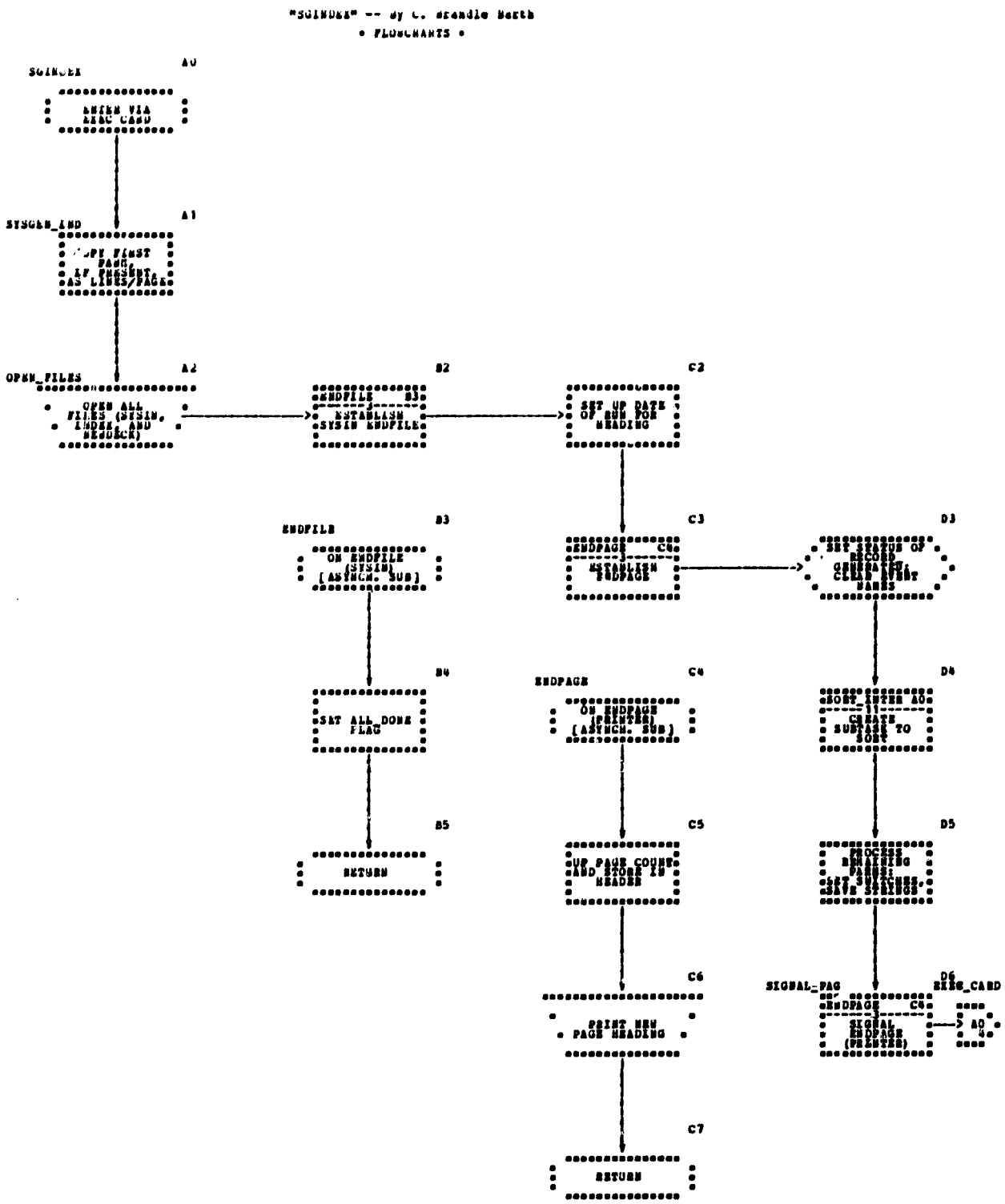


```
START LEVEL TEST

556     2       PRINT_THUMB_TAB:  PROCEDURE;    /*********************************/01438000
                                                /* THIS BLOCK PRINTS THE THUMB TABS*/01439000
                                                /*    FOR THE FIRST AND LAST INDEX */01440000
                                                /*    ITEMS ON THE PAGE AT THE     */01441000
                                                /*    BOTTOM OF THE PAGE.          */01442000
                                                /*********************************/01443000

557     1           SORT_RECORD_AREA = SORTED_ENTRY (1);                        01444000
                                                /* PUT FIRST ENTRY INTO STRUCTURE. */01445000
558     1       PUT FILE (PRINTER)              /* PRINT THE THUMB TABS.           */01446000
                    EDIT (SORT_RECORD.ITEM, SORT_RECORD.ITEM,                   01447000
                    RIGHT_RECORD.ITEM, RIGHT_RECORD.ITEM)                       01448000
                    (LINE(THUMB_SPOT), COLUMN(113), A(8),                       01449000
                    SKIP(0), COLUMN(114), A(8),                                 01450000
                    SKIP(1), COLUMN(113), A(8),                                 01451000
                    SKIP(0), COLUMN(113), A(8));                                01452000
559     1       END PRINT_THUMB_TAB;            /*********************************/01453000


                                                                                01454000

560     2   END SORT_INTERFACE;                 /*********************************/01455000
```

```
      /* *                    * S G I N D E X *                    * */00001000

STMT LEVEL NEST

561    1        VERIFY: PROCEDURE (VERIFY_STRING, SUB_STRING) BINARY FIXED;          01456000

                                    /*******************************/01457000
                                    /* THIS BLOCK SIMULATES THE    */01458000
                                    /*    "VERIFY" FUNCTION WHICH IS*/01459000
                                    /*    A BUILTIN FUNCTION WITH   */01460000
                                    /*    PL/I VERSION 5 (RELEASE 19).*/01461000
                                    /*    IT IS INCLUDED HERE FOR   */01462000
                                    /*    RUNNING UNDER RELEASE 18 AND*/01463000
                                    /*    SHOULD BE REMOVED LATER TO*/01464000
                                    /*    USE THE BUILTIN VERSION.  */01465000
                                    /*******************************/01466000

562    2            DECLARE                                                          01467000
                         VERIFY_STRING CHAR (100) VARYING,                          01468000
                                    /* STRING TO BE SEARCHED FOR VALID */01469000
                                    /*    CHARACTERS.               */01470000
                         SUB_STRING CHAR (30) VARYING,                              01471000
                                    /* STRING CONTAINING VALID CHAR- */01472000
                                    /*    ACTERS CONSIDERED ACCEPTABLE*/01473000
                                    /*    IN "VERIFY_STRING".       */01474000
                         CHARACTER CHAR (1) STATIC,                                 01475000
                                    /* WORK AREA.                   */01476000
                         (N, M) BINARY FIXED STATIC;                                01477000
                                    /* WORK VARIABLES;  N IS LENGTH */01478000
                                    /*    OF "VERIFY_STRING";  M IS */01479000
                                    /*    THE CHARACTER OF          */01480000
                                    /*    "VERIFY_STRING" CURRENTLY */01481000
                                    /*    BEING PROCESSED.          */01482000

563    2            M = 1;                  /* WE WILL START WITH THE FIRST  */01483000
                                    /*    CHARACTER IN "VERIFY_STRING".*/01484000
564    2            N = LENGTH (VERIFY_STRING);                                      01485000
                                    /* SAVE STRING LENGTH TO TELL WHEN */01486000
                                    /*    WE'VE FAILED.             */01487000

565    2        LOOP:                                                                01488000
                    IF M > N                /* IF WE'RE DONE (OR N = 0),     */01489000
566    2                THEN RETURN (0);    /*    SEARCH FAILED;  RESULT = 0.*/01490000
567    2            CHARACTER = SUBSTR (VERIFY_STRING, M, 1);                        01491000
                                    /* GO GET THE NEXT CHARACTER.    */01492000
568    2            IF INDEX (SUB_STRING, CHARACTER) = 0                             01493000
569    2                THEN RETURN (M);    /* IF THE CHARACTER IS MISSING IS*/01494000
                                    /*    THE LIST OF VALID CHARACTERS,*/01495000
                                    /*    RETURN ITS POSITION.      */01496000
570    2            M = M + 1;              /* ELSE GO LOOK AT THE NEXT ONE. */01497000
571    2            GO TO LOOP;                                                      01498000

      /* *                    * S G I N D E X *                    * */00001000

STMT LEVEL NEST

572    2            END VERIFY;             /*******************************/01499000

                                                                                    01500000

                                                                                    01501000

573    1        END SGINDEX;                /*******************************/01502000
```

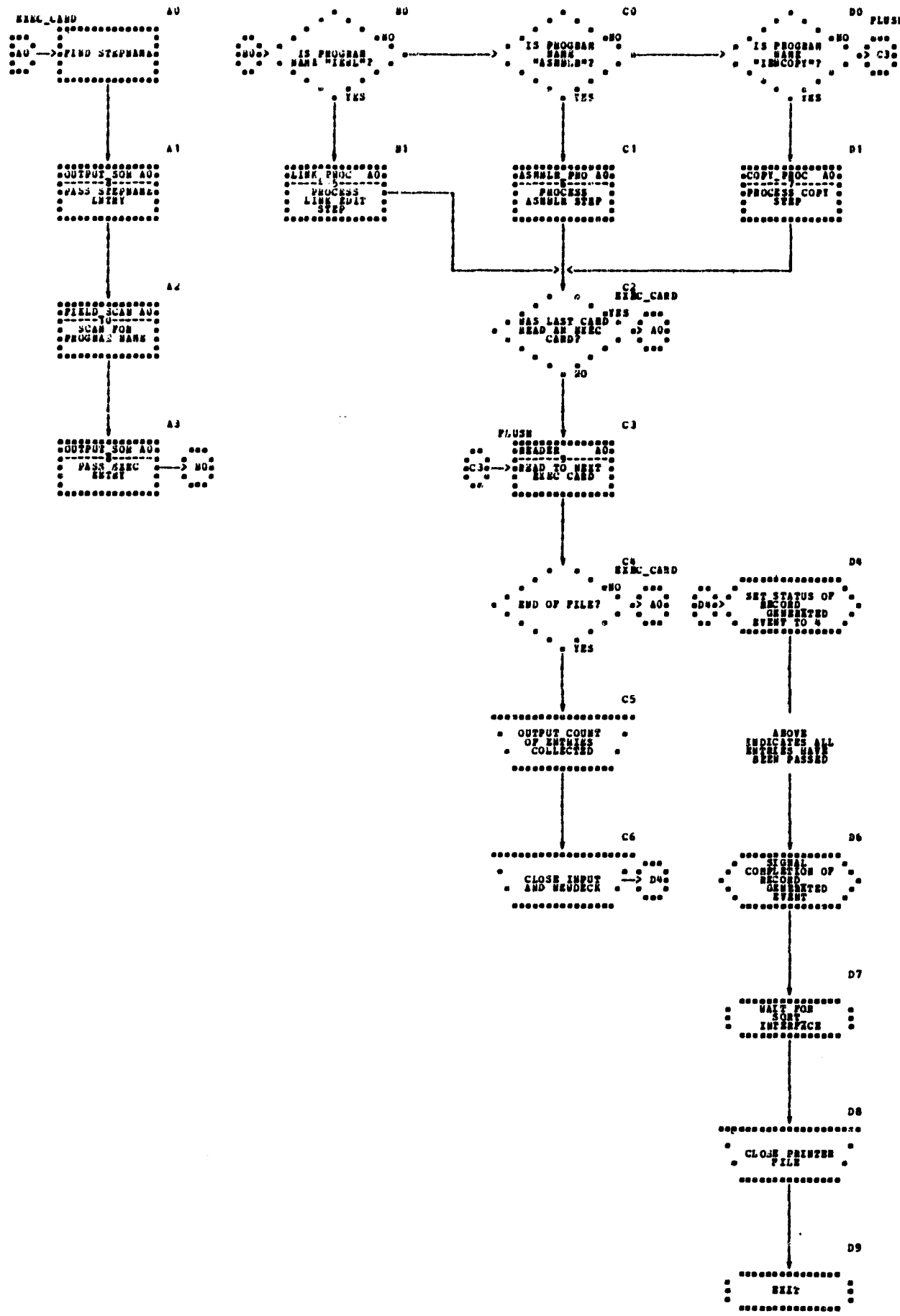
APPENDIX C

FLOW CHARTS



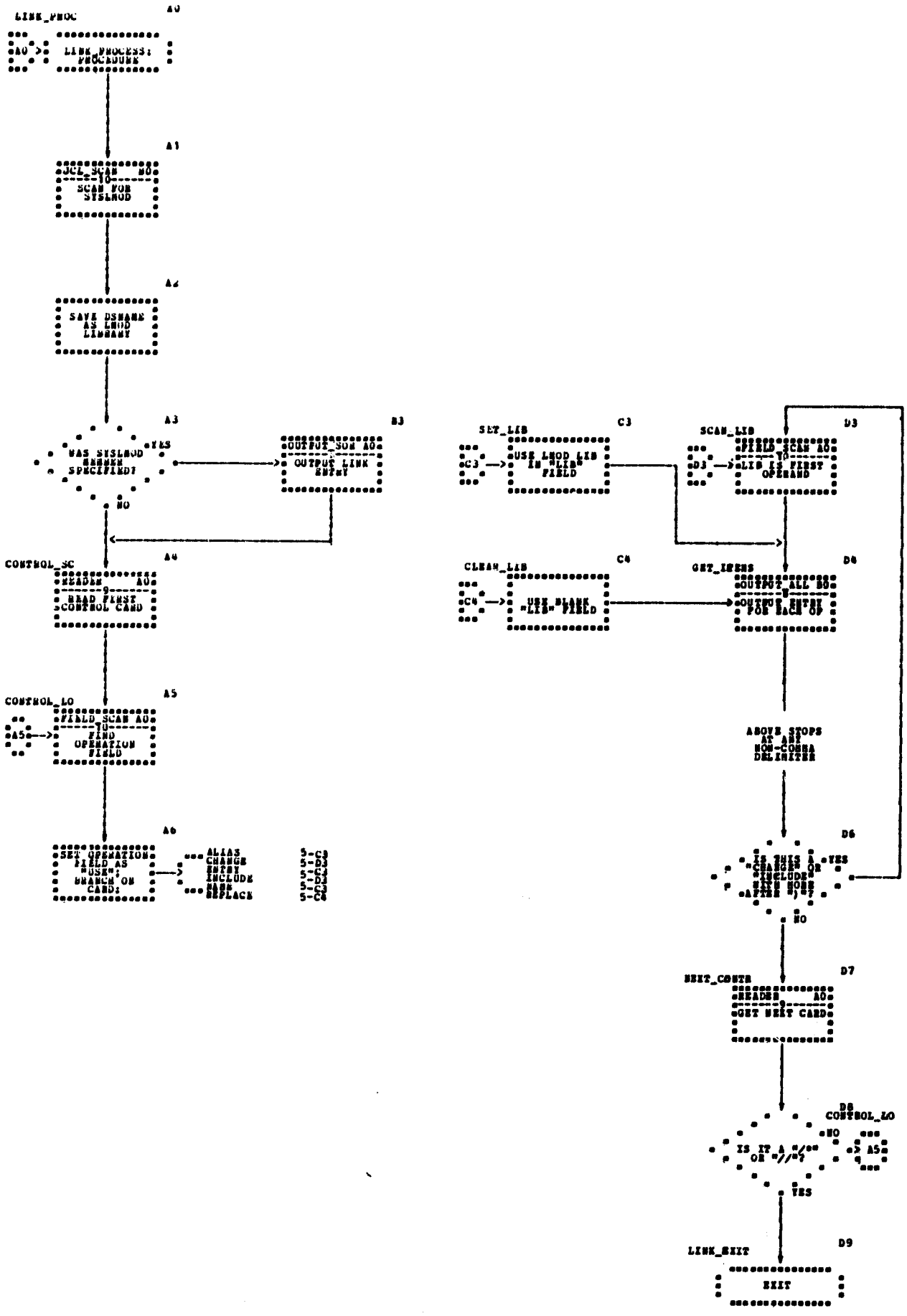
"SGINDEX" -- by C. Brandis Barth

• FLOWCHARTS •

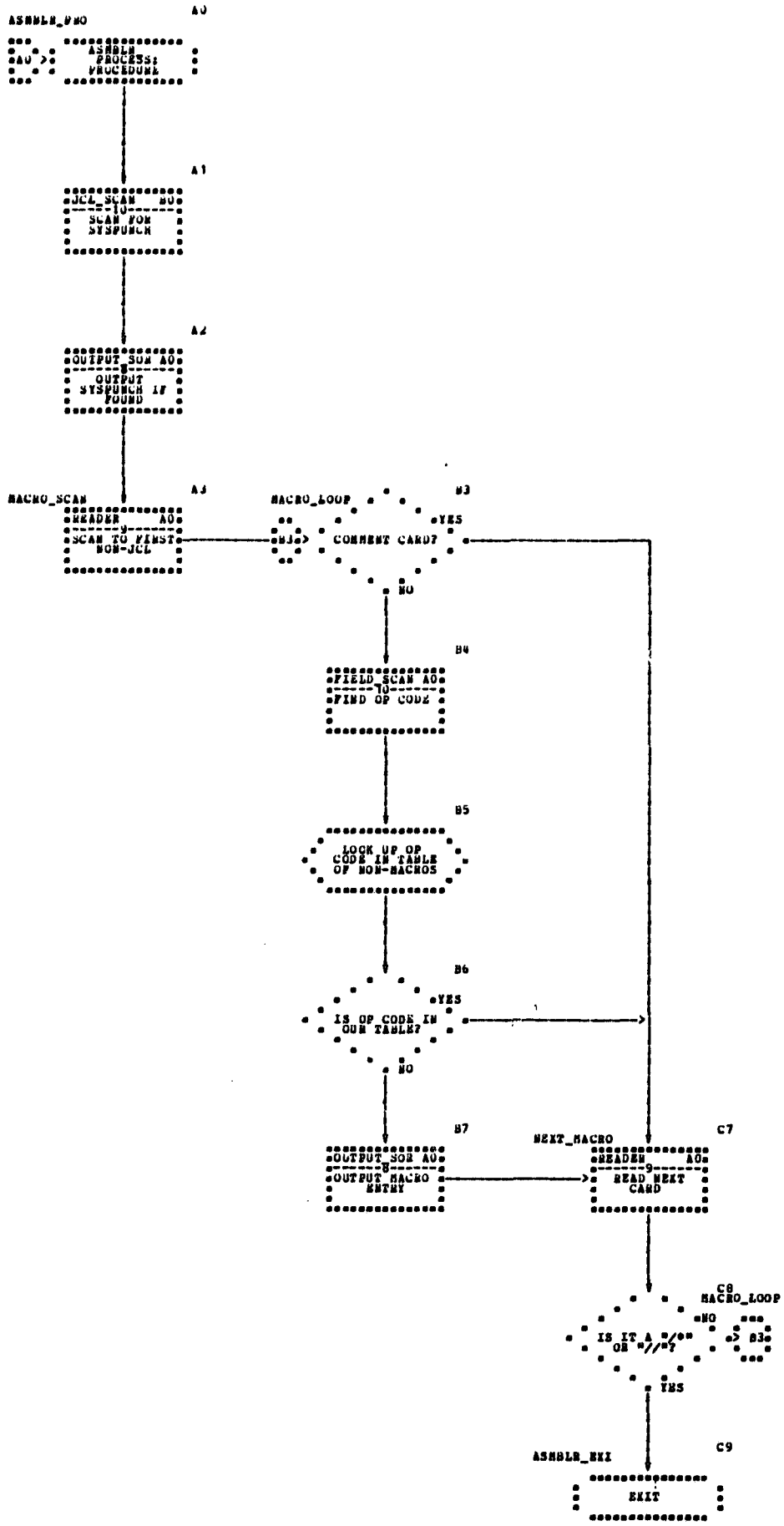


"SCINDEX" -- By C. Wrandle Booth

• FLOWCHARTS •

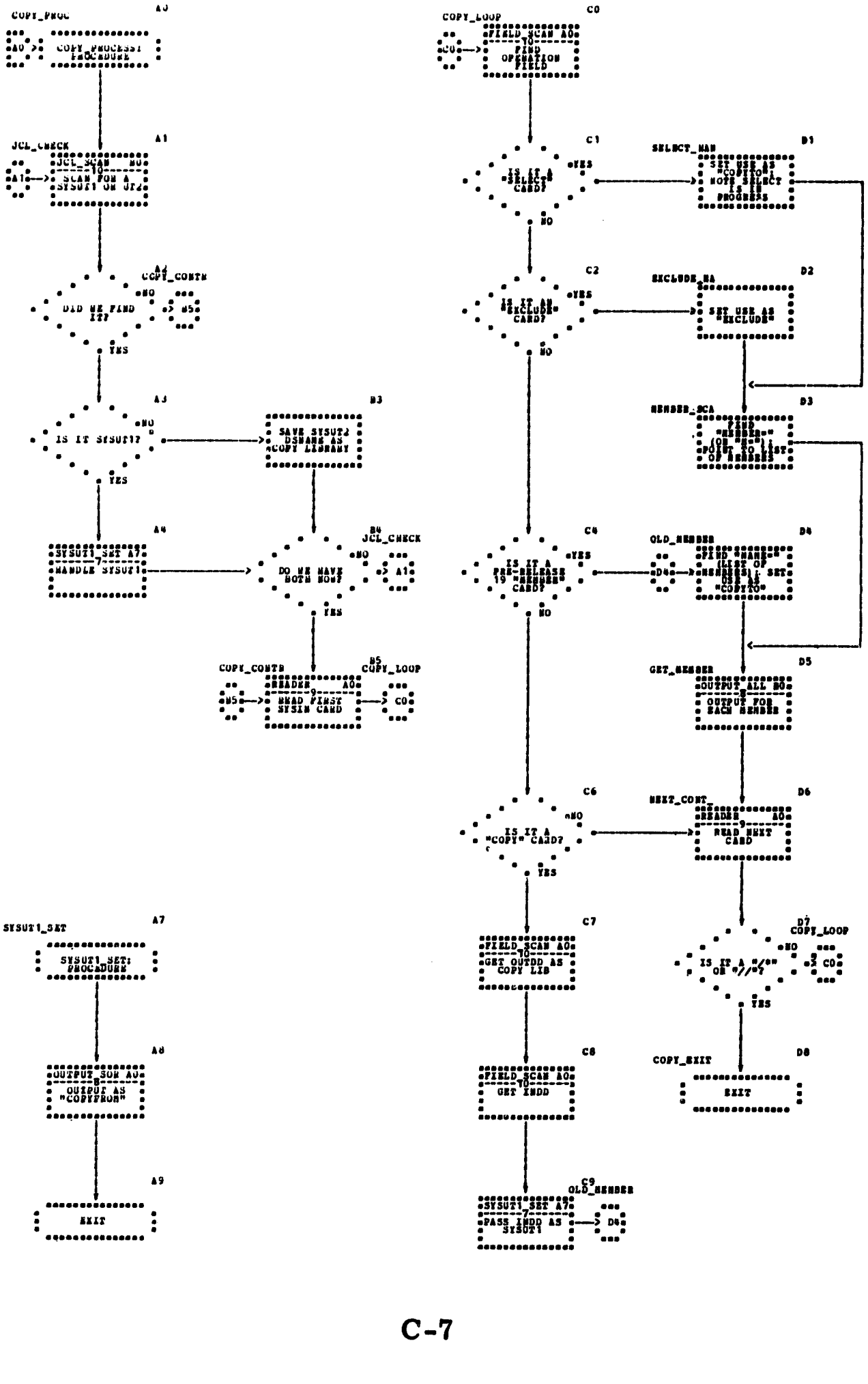


"JOINDEX" -- by C. Brandis Marth
• FLOWCHARTS •



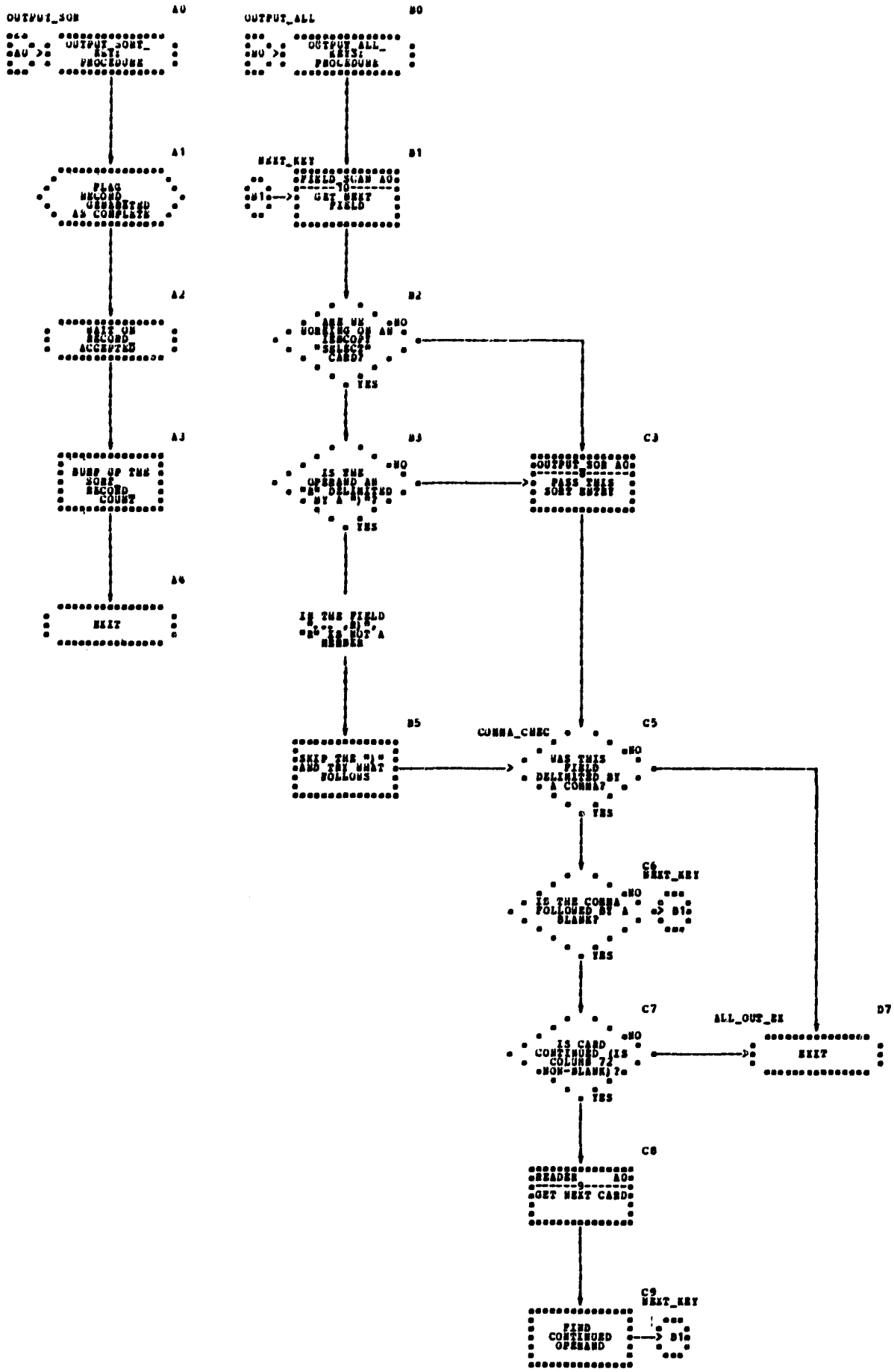
"SCRIBER" -- By C. Brandie Barth

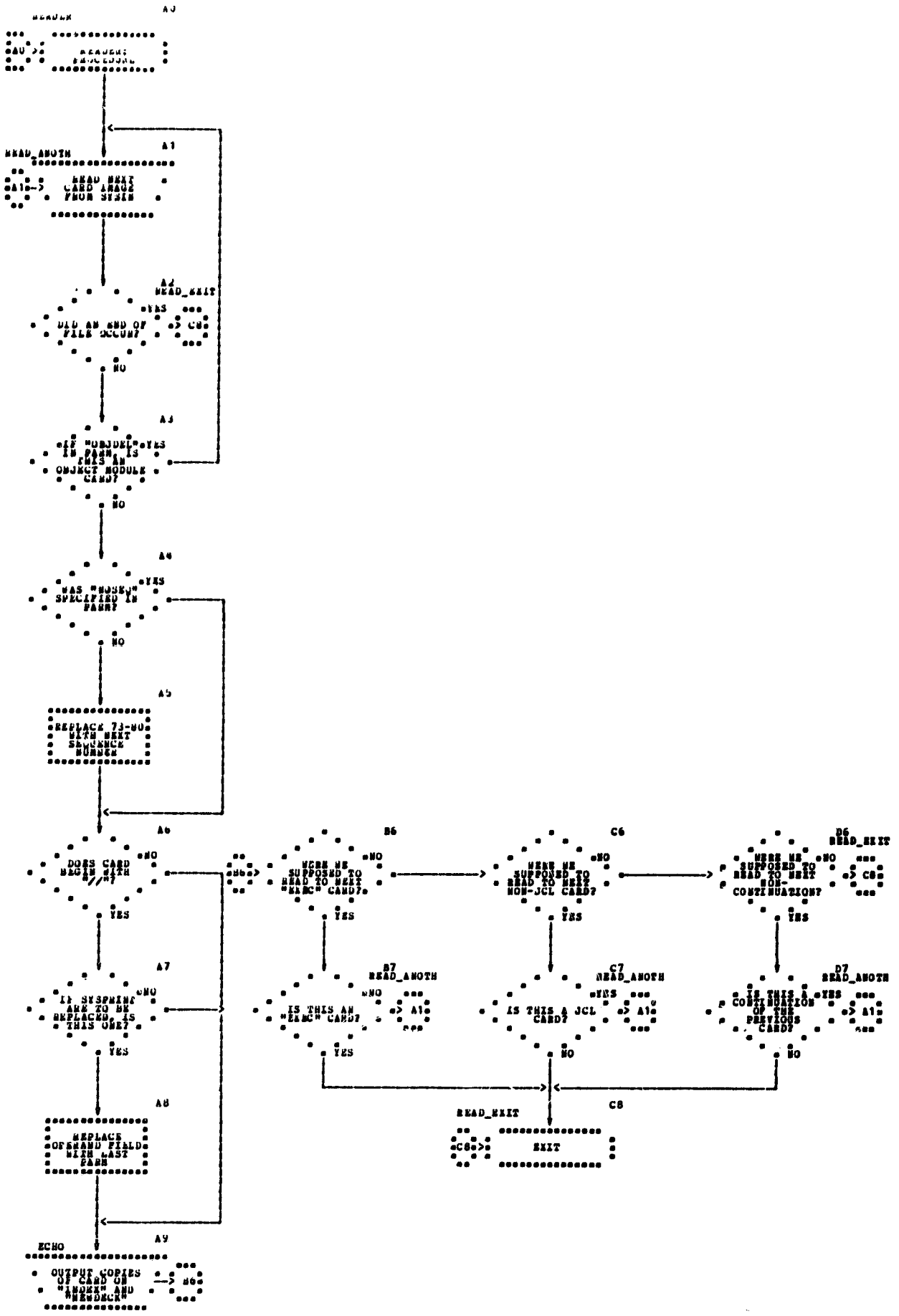
• FLOWCHARTS •

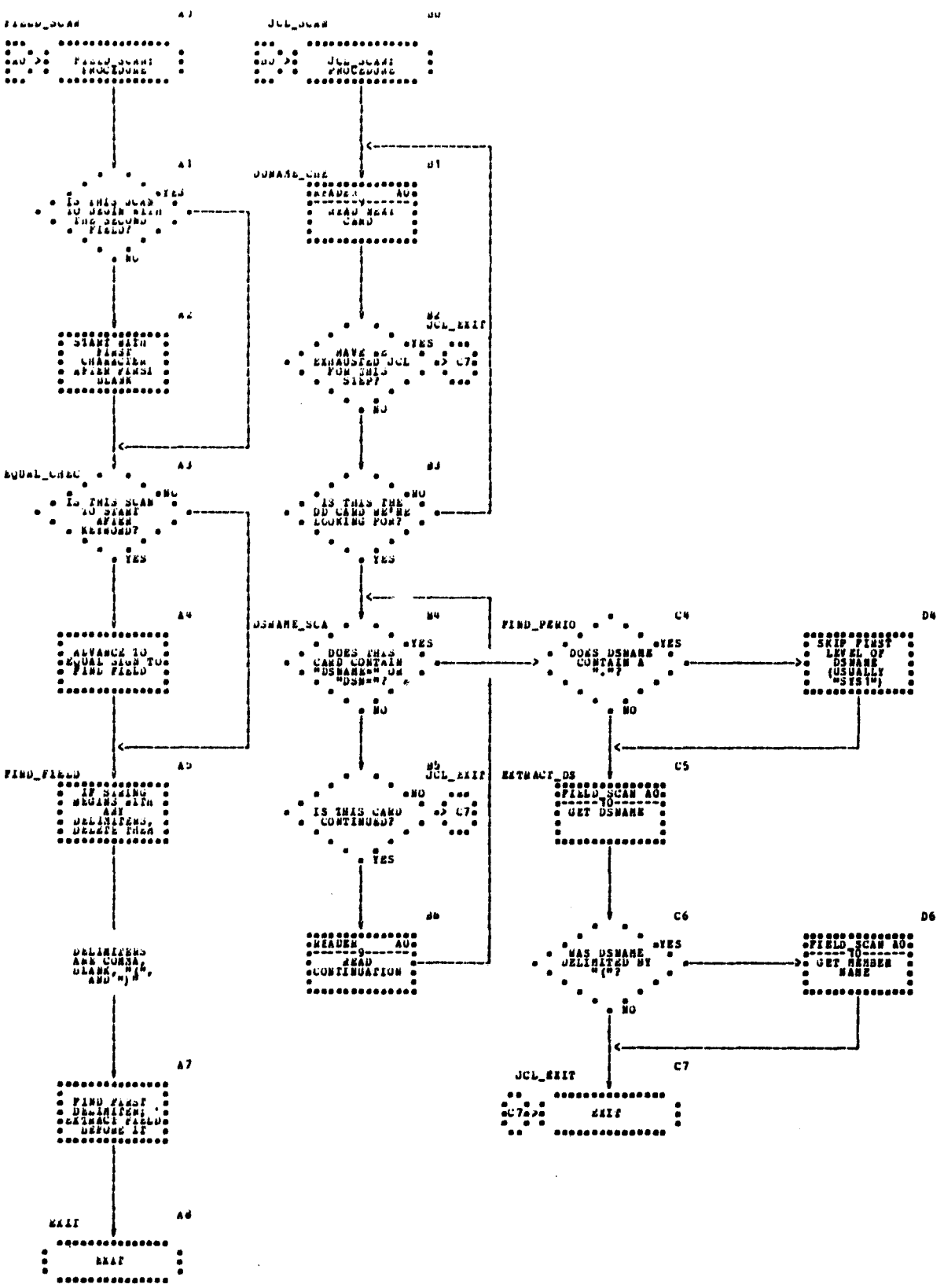


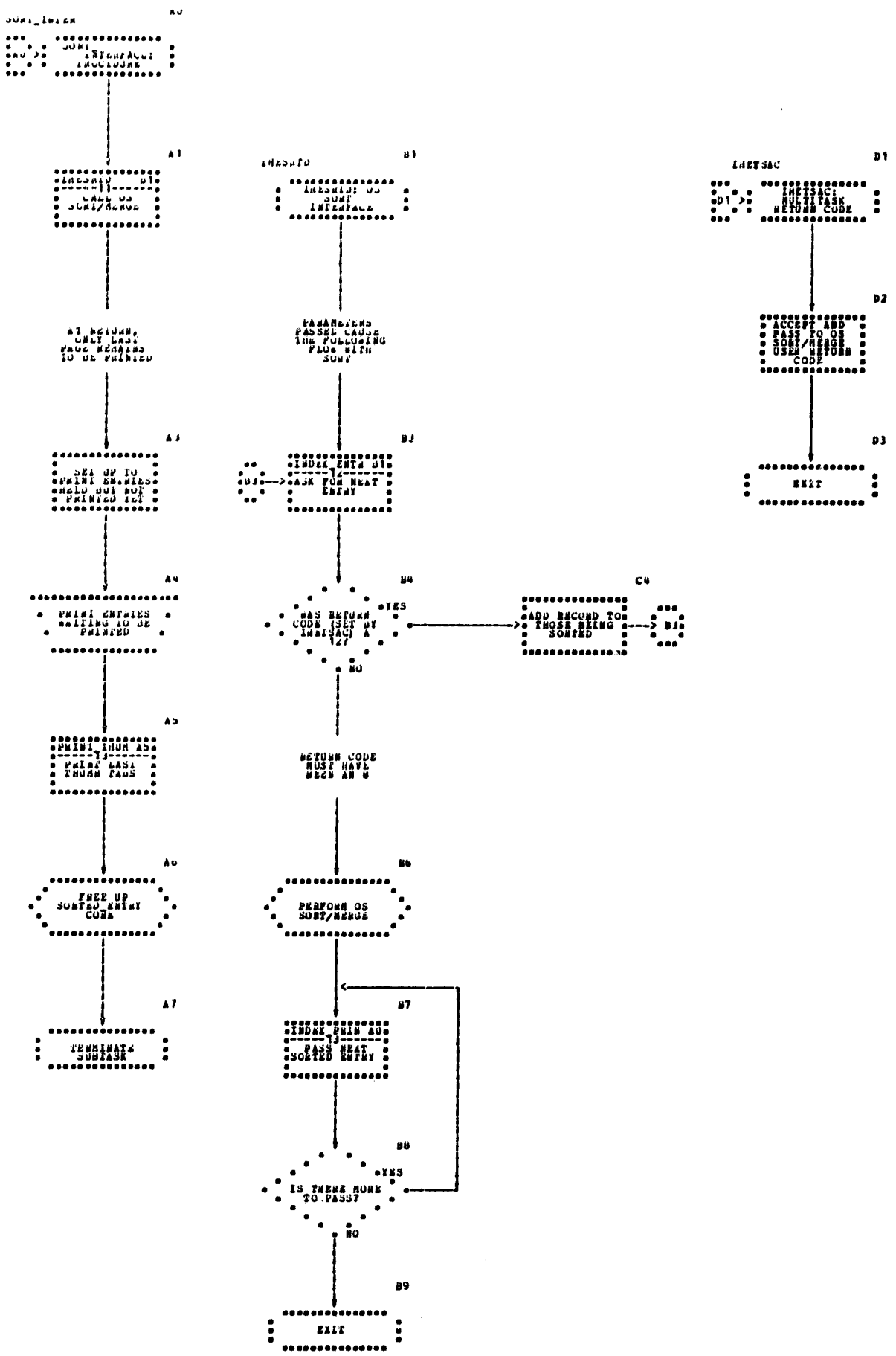
"SUITEDEX" -- by C. Brandis Smith

• FLOWCHARTS •

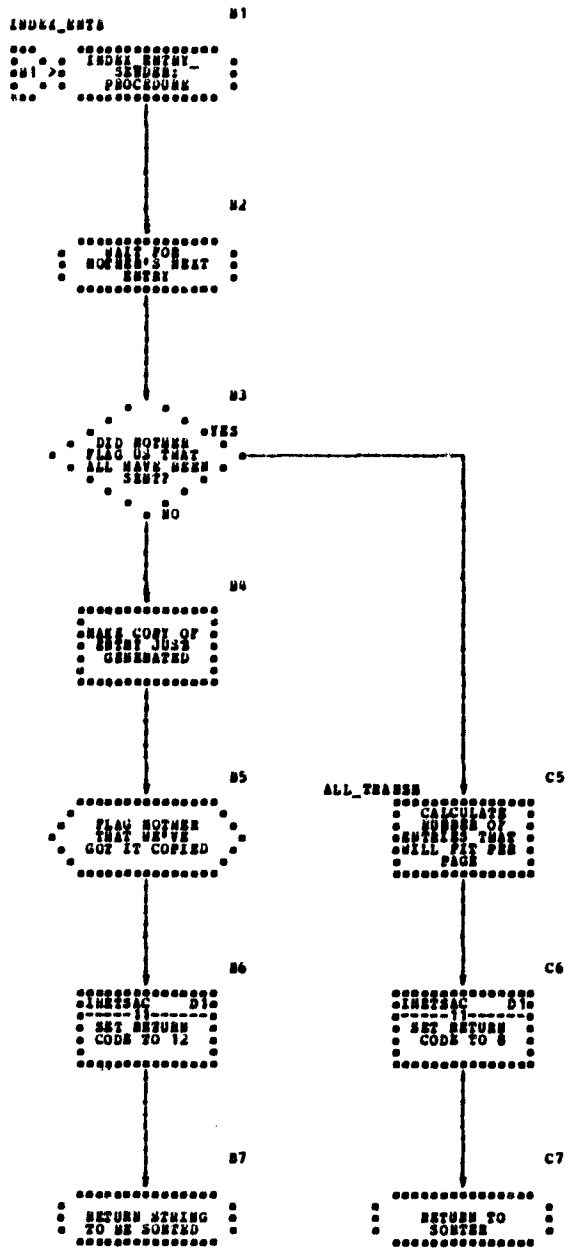



READER
A0
READER
EXECUTION
READ_ANOTH
A1
READ NEXT
CARD AREA2
FROM SYSIN
A2
READ_EXIT
YES
DID AN END OF
FILE OCCUR?
C8
NO
A3
IF "SUBJOB" YES
IN PARM IS
THIS AN
OBJECT MODULE
CARD?
NO
A4
WAS "NOSEQ"
SPECIFIED IN
PARM?
YES
NO
A5
REPLACE 73-80
WITH NEXT
SEQUENCE
NUMBER
A6
DOES CARD
BEGIN WITH
"//"?
NO
YES
A7
IS SYSPRINT
ARE TO BE
REPLACED, IS
THIS ONE?
NO
YES
A8
REPLACE
OPERAND FIELD
WITH LAST
PARM
ECHO
A9
OUTPUT COPIES
OF CARD ON
"INDEX" AND
"RENDECK"
B6
WERE WE
SUPPOSED TO
READ TO NEXT
"SYSIN" CARD?
NO
YES
B7
READ_ANOTH
IS THIS AN
"EXEC" CARD?
NO
A1
YES
C6
WERE WE
SUPPOSED TO
READ TO NEXT
NON-JCL CARD?
NO
YES
C7
READ_ANOTH
IS THIS A JCL
CARD?
YES
A1
NO
D6
READ_EXIT
WERE WE
SUPPOSED TO
READ TO NEXT
NON-
CONTINUATION?
NO
C8
YES
D7
READ_ANOTH
IS THIS A
CONTINUATION
OF THE
PREVIOUS
CARD?
YES
A1
NO
READ_EXIT
C8
EXIT


FIELD_SCAN
A1
FIELD_SCAN:
PROCEDURE
Is this SCAN to begin with the RECORD FIELD?
YES
NO
A2
START WITH FIRST CHARACTER after FIRST BLANK
EQUAL_CHEC
A3
Is this SCAN to START after KEYWORD?
YES
A4
ADVANCE TO EQUAL SIGN TO FIND FIELD
FIND_FIELD
A5
IF STRING BEGINS with ANY DELIMITERS, DELETE THEM
DELIMITERS ARE COMMA, BLANK, "(", AND ")"
A7
FIND FIRST DELIMITER; EXTRACT FIELD BEFORE IT
EXIT
A8
EXIT
JCL_SCAN
B0
JCL_SCAN:
PROCEDURE
DSNAME_CHK
B1
READER AUG
READ NEXT CARD
B2
JCL_EXIT
HAVE WE EXHAUSTED JCL FROM THIS STEP?
YES
-> C7
NO
B3
IS THIS THE DD CARD WE'RE LOOKING FOR?
YES
NO
DSNAME_SCA
B4
DOES THIS CARD CONTAIN "DSNAME=" OR "DSN="?
YES
NO
B5
JCL_EXIT
IS THIS CARD CONTINUED?
YES
NO
-> C7
B6
READER AUG
READ CONTINUATION
FIND_PERIO
C4
DOES DSNAME CONTAIN A "."?
YES
NO
D4
SKIP FIRST LEVEL OF DSNAME (USUALLY "SYS1")
EXTRACT_DS
C5
FIELD_SCAN AUG
GET DSNAME
C6
WAS DSNAME DELIMITED BY "("?
YES
NO
D6
FIELD_SCAN AUG
GET MEMBER NAME
JCL_EXIT
C7
C7->
EXIT

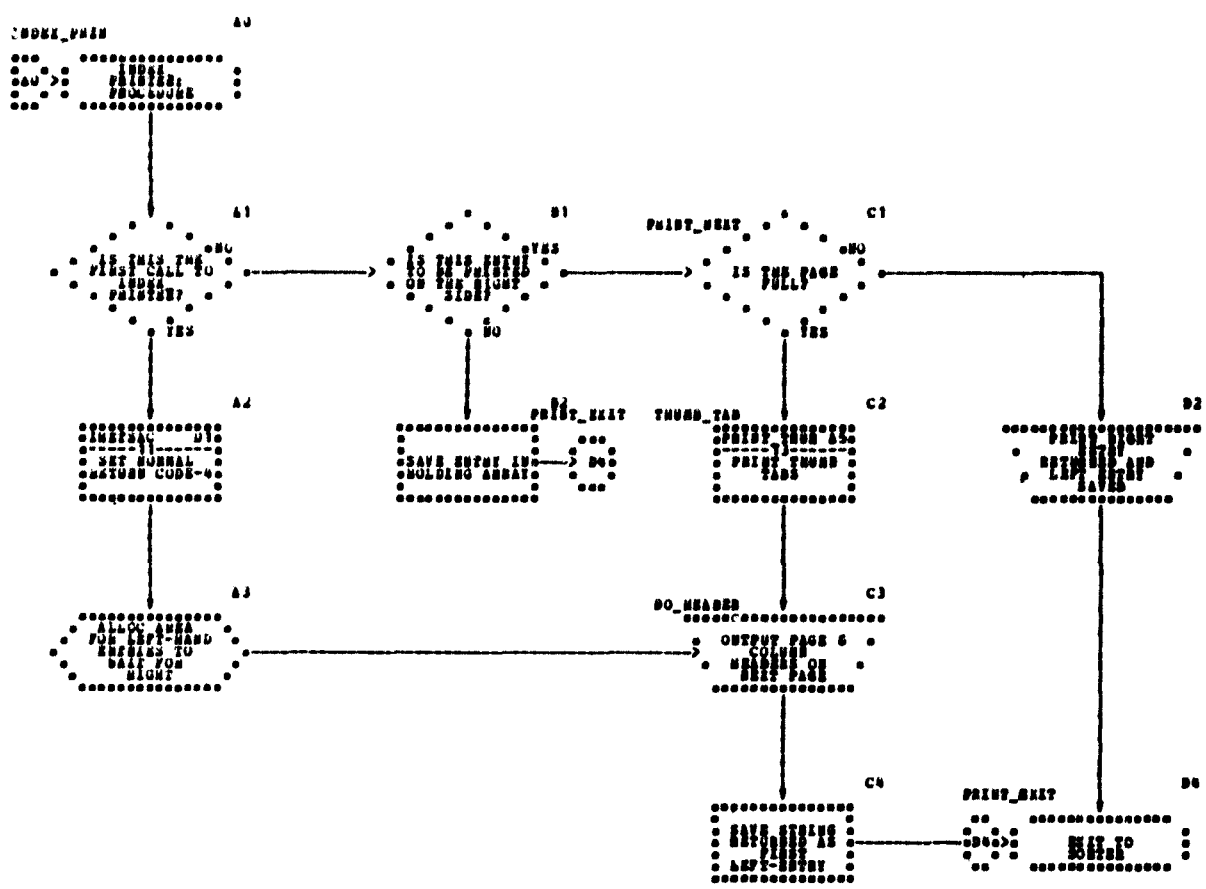

SORT_INTER
A0
SORT INTERFACE INSTRUCTIONS
A1
INTERNAL SORT CALL SORT/MERGE
A1 RETURN, ONLY LAST PASS REMAINS TO BE PRINTED
A3
SEE IF TO PRINT ENTRIES HELD OUT NOT PRINTED YET
A4
PRINT ENTRIES WAITING TO BE PRINTED
A5
PRINT INUM AS PRINT LAST THOSE PASS
A6
FREE UP SORTED_ENTRY CORE
A7
TERMINATE SUBTASK
INTERFC
B1
INTERNAL SORT INTERFACE
PARAMETERS PASSED CAUSE THE FOLLOWING FLOW WITH SORT
B3
INDEX_ENTRY B1 ASK FOR NEXT ENTRY
B4
WAS RETURN CODE (SET BY INSERTAC) A 'Z'?
YES
NO
ADD RECORD TO THOSE BEING SORTED
B5
RETURN CODE MUST HAVE BEEN AN 8
B6
PERFORM OS SORT/MERGE
B7
INDEX_PNUM A0 PASS NEXT SORTED ENTRY
B8
IS THERE MORE TO PASS?
YES
NO
B9
EXIT
INSTSAC
D1
INSTSAC1 MULTITASK RETURN CODE
D2
ACCEPT AND PASS TO OS SORT/MERGE USER RETURN CODE
D3
EXIT
C4

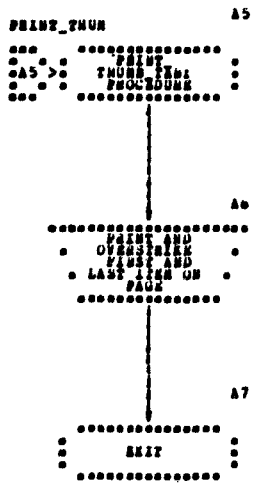
"SORTER" -- by C. Gradle Beith
• SUBTASK FLOWCHART •



"SOLDOEX" -- by C. Brandis Bacon

• SUBTASK FLOWCHART •





"SGINDEX" — By C. Wrandle Barth

• Flowchart Cross Reference •

page-coordinate	flowchart name	PL/I name
8-D7	ALL–OUT–FX	ALL–OUT–EXIT
12-C5	ALL–TRANSM	ALL–TRANSMITTED
6-C9	ASMBLR–EXI	ASMBLR–EXIT
6-A0	ASMBLR–PRO	ASMBLR–PROCESS
5-C4	CLEAR–LI B	CLEAR–LIB
8-C5	COMMA–CHEC	COMMA–CHECK
5-A5	CONTROL–LO	CONTROL–LOOP
5-A4	CONTROL–SC	CONTROL–SCAN
7-B5	COPY–CONTR	COPY–CONTROL–CARDS
7-D8	COPY–EXIT	COPY–EXIT
7-C0	COPY–LOOP	COPY–LOOP
7-A0	COPY–PROC	COPY–PROCESS
10-B1	DDNAME–CHE	DDNAME–CHECK
13-C3	DO–HEADER	DO–HEADER
10-B4	DSNAME–SCA	DSNAME–SCAN
9-A9	ECHO	ECHO
3-B3	ENDFILE	- - -
3-C4	ENDPAGE	- - -
10-A3	EQUAL–CHEC	EQUAL–CHECK
7-D2	EXCLUDE–HA	EXCLUDE–HANDLER
4-A0	EXEC–CARD	EXEC–CARD–PROCESS
10-A8	EXIT	EXIT
10-C5	EXTRACT–DS	EXTRACT–DSNAME
10-A0	FIELD–SCAN	FIELD–SCAN
10-C4	FIND–PERIO	FIND–PERIOD
4-C3	FLUSH	FLUSH
5-D4	GET–ITEMS	GET–ITEMS
7-D5	GET–MEMBER	GET–MEMBERS
11-B1	IHESRTD	IHESRTD
11-D1	IHETSAC	IHETSAC
12-B1	INDEX–ENTR	INDEX–ENTRY–SENDER
13-A0	INDEX–PRIN	INDEX–PRINTED
7-A1	JCL–CHECK	JCL–CHECK
10-C7	JCL–EXIT	JCL–EXIT
10-B0	JCL–SCAN	JCL–SCAN
5-D9	LINK–EXIT	LINK–EXIT
5-A0	LINK–PROC	LINK–PROCESS
6-B3	MACRO–LOOP	MACRO–LOOP
6-A3	MACRO–SCAN	MACRO–SCAN
7-D3	MEMBER–SCA	MEMBER–SCAN
7-D6	NEXT–CONT	NEXT–CONTROL–CARD
5-D7	NEXT–CONTR	NEXT–CONTROL
8-B1	NEXT–KEY	NEXT–KEY
6-C7	NEXT–MACRO	NEXT–MACRO
7-D4	OLD–MEMBER	OLD–MEMBER–CARDS
3-A2	OPEN–FILES	OPEN–FILES
8-B0	OUTPUT–ALL	OUTPUT–ALL–KEYS
8-A0	OUTPUT–SOR	OUTPUT–SORT–KEY
13-D4	PRINT–EXIT	PRINT–EXIT
13-C1	PRINT–NEXT	PRINT–NEXT–LINE
13-A5	PRINT–THUM	PRINT–THUMB–TAB
9-A1	READ–ANOTH	READ–ANOTHER
9-C8	READ–EXIT	READ–EXIT
9-A0	READER	READER
5-D3	SCAN–LIB	SCAN–LIB
7-D1	SELECT–HAN	SELECT–HANDLER
5-C3	SET–LIB	SET–LIB
3-A0	SGINDEX	SGINDEX
3-D6	SIGNAL–PAG	SIGNAL–PAGE
11-A0	SORT–INTER	SORT–INTERFACE
3-A1	SYSGEN–IND	SYSGEN–INDEX–CREATION
7-A7	SYSUT1–SET	SYSUT1–SET
13-C2	THUMB–TAB	THUMB–TAB